IL GRANDE LIBRO DEI GIOCHI

46 programmi per il Commodore 64

Franz Ende



EDIZIONE ITALIANA

GRUPPO EDITORIALE JACKSON

IL GRANDE LIBRO DEI GIOCHI

Franz Ende

46 Programmi per il Commodore 64

Introduzione alla grafica e al suono
Come programmare i Vostri Giochi
Animazioni
Suggerimenti e trucchi
Soubroutine di utilità

GRUPPO
EDITORIALE
JACKSON
Via Rosellini, 12
20124 Milano

Titolo originale: The Great Book of Games

TRADUZIONE: Pier Luigi Rossi
COPERTINA: Marcello Longhini
GRAFICA E IMPAGINAZIONE: Cristina De Venezia
SUPERVISIONE TECNICA: Daria Gianni
FOTOCOMPOSIZIONE: CorpoNove s.n.c. - Bergamo
STAMPA: Alberto Matarelli S.p.A. - Stabilimento Grafico

SOMMARIO

Prefazione V
Slot - machine 1
Roulette 5
Topi 9
Super tre 11
Lotto 15
Collisione 17
Flip - flop 19
Labirinto 23
Attacco 25
Artiglieria 29
Quadrato magico 33
Girotondo 35
Distanza tra due città 37
Curve 41
Zanzare 43
Tempo nel mondo 45
Concentrazione 49
Danza delle palle 51
Space shuttle 53
Introduzione ai numeri binari 55
Caduta libera e lancio verso l'alto 59
Il movimento obliquo 61
Grafica con il COMMODORE 64 63
Grafica in alta risoluzione 71
La funzione multicolore 77
Scrolling regolare 79
Sprite 83
Generatore di sprite 87
Suono 95
Joystick e paddle 103

Non è facile identificare i caratteri di controllo in un listato BASIC del COMMODORE 64: elenchiamo qui sotto i simboli più usati.

ASCII	PRINTS	DESCRIPTION	
146		Reverse Off	Usa il tasto CTRL
18		Reverse On	Usa il tasto CTRL
3		RUN/STOP	
147		Clear	CRL
19		Home	
145		Cursor Up	
17		Cursor Down	
157		Cursor Left	
29		Cursor Right	
148		Insert	

PREFAZIONE

Il COMMODORE 64 rappresenta un eccellente investimento in relazione al suo costo.

Le caratteristiche che fanno emergere questo computer rispetto ai concorrenti comprendono: un'ampia memoria standard di 64 K byte di RAM, eccellenti possibilità di grafica e di suono e in più una delle migliori tastiere disponibili sul mercato.

Il computer è basato sulla CPU 6510, che è una versione evoluta della ben nota 6502.

La prima parte di questo libro contiene una serie di programmi di giochi che sono pronti per essere battuti ed essere messi in funzione con il comando RUN.

Tutti questi programmi sono scritti in semplice BASIC COMMODORE.

La seconda parte del libro vuole insegnarvi come sfruttare la potenza delle nuove caratteristiche del vostro COMMODORE 64, per esempio il colore, la grafica ad alta risoluzione, lo scrolling, gli sprite, il suono, i joystick e le paddle.

Quando avrete terminato la lettura della seconda parte del libro sarete capaci di scrivere i vostri programmi, anche i più sofisticati o completerete quelli della prima parte aggiungendo colore, suono e altre caratteristiche.

SLOT - MACHINE

Questo programma simula una slot - machine.

All'inizio dovete depositare una certa somma di denaro indicandone l'ammontare (per esempio 1, 2 o 5). Il programma indicherà ogni volta l'ammontare del vostro credito in quel momento. Numeri alternati vengono mostrati in una finestra sullo schermo e per fermarli dovete premere la barra spaziatrice.

A seconda della combinazione dei numeri mostrati potrete vincere una certa quantità di denaro che si aggiunge al vostro credito.

Se non vincete, 20 cent per partita saranno sottratti dal vostro credito.

```
99 REM 1. SLOT MACHINE
100 GOTO970
110 PRINT"❐
120 PRINT"▚▶▶▶▶▶▶▶▶▶▶▶▶╭──────────────────────╮
130 PRINT"▶▶▶▶▶▶▶▶▶▶▶▶▶|______________________|"
140 PRINT"▶▶▶▶▶▶▶▶▶▶▶▶▶|▚   C64-LUCK    ▀|"
150 FORZ=1TO4:PRINT"▶▶▶▶▶▶▶▶▶▶▶▶▶|▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶|"
160 NEXTZ
170 PRINT"▶▶▶▶▶▶▶▶▶▶▶▶▶| *   \ | /   *  |"
180 PRINT"▶▶▶▶▶▶▶▶▶▶▶▶▶| *    \|/    *  |"
190 PRINT"▶▶▶▶▶▶▶▶▶▶▶▶▶| *  ╭──────╮ *  |"
200 PRINT"▶▶▶▶▶▶▶▶▶▶▶▶▶|  *|      |*   |"
210 PRINT"▶▶▶▶▶▶▶▶▶▶▶▶▶| *  ╰──────╯ *  |"
220 PRINT"▶▶▶▶▶▶▶▶▶▶▶▶▶| *     ↑↑    *  |"
230 PRINT"▶▶▶▶▶▶▶▶▶▶▶▶▶|  * CREDIT *    |"
240 PRINT"▶▶▶▶▶▶▶▶▶▶▶▶▶|                |"
250 PRINT"▶▶▶▶▶▶▶▶▶▶▶▶▶|    ━━ ━━ ━━    |"
260 PRINT"▶▶▶▶STOP▶▶▶|<──── STOP       |"
270 PRINT"▶▶▶▶PREMI▶▶|      ┌─────┐     |"
280 PRINT"▶▶<SPAZIO>▶|      └─────┘     |"
290 PRINT"▶▶▶▶▶▶▶▶▶▶▶▶▶╰────────────────╯
300 PRINT"▚▼▼▼▼▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶╭───╮▶▶╭───╮▶▶╭───╮
310 PRINT"▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶|00|▶▶|00|▶▶|00| ▮
320 PRINT"▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶╰───╯▶▶╰───╯▶▶╰───╯
330 REM
340 GOSUB880
350 IFG=0THEN790
360 FORX=1TO20
370 GETQ$:IFQ$=" "ANDX>2THENX=20:GOTO570
380 IFX=2THENG=G-20:GOSUB880
390 A=INT(RND(1)*10)*10:POKE1240,87:POKE55512,4:IF(A/20)<>INT(A/20)THEN390
400 POKE1245,81:POKE55517,4
410 B=INT(RND(1)*10)*10:POKE1245,87:POKE55517,4:IF(B/20)<>INT(B/20)THEN410
420 POKE1240,81:POKE55512,4
430 C=INT(RND(1)*10)*10:POKE1240,87:POKE55512,4:IF(C/20)<>INT(C/20)THEN430
440 POKE1245,81:POKE55517,4
450 IFA=0THENA$="**":GOTO470
```

```
460 A$=RIGHT$(STR$(A),2)
470 IFB=0THENB$="**":GOTO490
480 B$=RIGHT$(STR$(B),2)
490 IFC=0THENC$="**":GOTO510
500 C$=RIGHT$(STR$(C),2)
510 REM
520 PRINT"";
530 IFA=BANDA=CTHENPRINT""A$""B$""C$"":GOTO570
540 IFA<>CANDB=0THENPRINTA$""B$""C$:GOTO570
550 IFA=CANDB=0THENPRINT""A$""B$""C$"":GOTO570
560 PRINTA$""B$""C$
570 NEXTX
580 POKE1240,102:POKE1245,102:POKE55512,4:POKE55517,4
590 IFA=0ANDB=0ANDC=0THENG=G+5000:GOSUB118:GOSUB125:GOTO670
600 IFA=CANDB=0THENG=G+A:GOTO670
610 IFB=0ANDA<>CTHENG=G+20:GOTO670
620 IFA=BANDA=CTHENG=G+A:GOTO670
630 IFA=BANDB<>CTHEN670
640 IFA=CANDB<>CTHEN670
650 IFB=CANDA<>BTHEN670
660 IFA<>BANDA<>CTHENG=G+0
670 GOSUB880
680 IFG<20THEN790
690 PRINT"";:FORI=1TO20:PRINT"";:NEXT:FORI=1TO17:PRINT"";:NEXT
700 PRINT"    "
710 FORI=1TO18:PRINT"";:NEXT:PRINT""
720 PRINT"              PREMI <SPAZIO>"
730 GETF$:IFF$<>" "THEN730
740 PRINT"";:FORI=1TO20:PRINT"";:NEXT:FORI=1TO17:PRINT"";:NEXT
750 PRINT"          "
760 FORI=1TO18:PRINT"";:NEXT:PRINT"   "
770 PRINT"                                 "
780 GOTO330
790 PRINT" ← DEPOSITA"
800 PRINT"UN GETTONE";
810 GETE$:IFE$=""ORVAL(E$)=0THEN810
820 E=VAL(E$):IFE>0THENPRINT"DOLLARI "E$:FORZ=1TO1000:NEXT
830 IFE=0ORE<>1ANDE<>2ANDE<>5THEN1380
840 P=E*100:FORZ=1TO1000:NEXT
850 PRINT"                    "
860 PRINT"                     "
861 PRINT"                     "
870 G=G+P:GOSUB880:GOTO330
880 PRINT"";:IFG>999990THENPRINT"PASS!":END
881 PRINT"";:IFG>999990THENPRINT"PASS!":END
890 IFG=0THENG$="000":GOTO920
900 IFG>0THENG$=STR$(G)
910 G$=RIGHT$(G$,LEN(G$)-1)
920 M$=G$
930 O$=LEFT$(M$,(LEN(G$)-2)):P$=RIGHT$(M$,2):G$=O$+"."+P$
940 IFLEN(G$)<6THENG$=" "+G$:GOTO940
950 IFLEN(G$)>6THENG$=LEFT$(G$,6)
960 PRINTCHR$(18)G$:RETURN
970 PRINTCHR$(147);CHR$(18)"        SLOT MACHINE C64-LUCK          "
980 PRINT"VUOI LEGGERE LE REGOLE ? (S/N)
990 GETI$:IFI$=""THEN990
1000 IFI$<>"S"ANDI$<>"N"THEN990
1010 IFI$="S"THEN1030
1020 GOTO110
1030 PRINT"";
1040 PRINT"IL PROGRAMMA SIMULA UNA SLOT MACHINE"
1050 PRINT
1060 PRINT"LA SOLA DIFFERENZA RISPETTO A  UNA VERA"
1070 PRINT"SLOT MACHINE E'CHE IL COMPUTER NON PAGA"
1080 PRINT"NESSUNA VINCITA : MA NON VI CHIEDE"
1090 PRINT"NEMMENO DI ESSERE PAGATO !"
1100 PRINT
1110 PRINT"IL COMPUTER ACCETTA GETTONI SIMBOLICI"
1120 PRINT"DA   1  ,   2  , O   5  DOLLARI."
1130 PRINT
1140 PRINT"PER OGNI GIOCATA IL COMPUTER TOGLIE"
1150 PRINT"20 CENTS DAL VOSTRO CREDITO."
```

```
1160 PRINT
1170 PRINT"PREMI <SPAZIO> PER CONTINUARE"
1180 GETN$:IFN$<>" "THEN1180
1190 PRINTCHR$(147)TAB(13)"LISTA DEI PREMI"
1200 PRINT"[illegible] ** ** **  = PRIMO PREMIO   50 DOLLARI"
1210 PRINT" 60 ** 60  = VINCI L'AMMONTARE INDICATO"
1220 PRINT" 40 ** 80  = 20 CENTS"
1230 PRINT" 80 80 80  = 80 CENTS"
1240 PRINT" 60 60 60  = 60 CENTS"
1250 PRINT" 40 40 40  = 40 CENTS"
1260 PRINT" 20 20 20  = 20 CENTS"
1270 PRINT" 20 20 60  = NESSUN PREMIO"
1280 PRINT" ** 60 40  = NESSUN PREMIO"
1290 PRINT" 20 40 **  = NESSUN PREMIO"
1300 PRINT" 80 80 **  = NESSUN PREMIO"
1310 PRINT:PRINT"LE STELLE AL CENTRO SONO IL JOLLY."
1320 PRINT"SE I NUMERI A DESTRA E A SINISTRA SONO"
1330 PRINT"UGUALI VINCI L'AMMONTARE INDICATO."
1340 PRINT
1350 PRINT"PREMI <SPAZIO> PER CONTINUARE"
1360 GETN$:IFN$<>" "THEN1360
1370 GOTO110
1380 IFE>0THENPRINT"[illegible]RIFIUTATO !"
1390 FORZ=1TO1500:NEXT
1400 PRINT"[illegible]                    "
1410 GOTO790

READY.
```

ROULETTE

Il programma seguente simula il gioco della roulette in accordo alle regole internazionali.

Potete scommettere su un singolo numero (da 1 a 36) o su una serie di numeri, come ad esempio tutti i numeri neri (premete "B") o tutti i numeri rossi ("R"), sui primi dodici numeri ("DI"), su tutti i numeri pari ("P"), sui primi diciotto numeri ("MA") e così via.

All'inizio dovete comunicare la somma di denaro con la quale intendete i-niziare le scommesse.

```
100 REM 2. ROULETTE
110 PRINT"⊐"
120 PRINT"              ◪ ROULETTE ■"
130 PRINT:PRINT
140 PRINT"SIMULAZIONE DEL GIOCO IN ACCORDO ALLE"
150 PRINT"REGOLE INTERNAZIONALI."
160 PRINT
170 PRINT"PUOI PUNTARE SU NUMERI DA"
180 PRINT"◪ 1-36 ■"
190 PRINT
200 PRINT"SONO ANCHE POSSIBILI ◪SERIE■, ◪COLONNE"
210 PRINT"◪DOZZINE■, ◪MANQUE■, ◪PASSE■"
220 PRINT"◪TRASVERSALI■, PER ESEMPIO :"
230 PRINT
240 PRINT"◪TA■=TRASVERSALE 1-2-3"
250 PRINT"◪TB■=TRASVERSALE 4-5-6"
260 PRINT"◪TM■=TRASVERSALE 34-35-36"
270 PRINT
280 PRINT"SE ESCE LO ◪ 0 ■ TUTTE LE SCOMMESSE"
290 PRINT"SONO PERSE ! ! !"
300 PRINT
310 PRINT"PREMI UN TASTO PER CONTINUARE"
320 GETA$:IFA$=""THEN320
330 PRINT"⊐"
340 PRINT
350 PRINT"◪PAIR■ (◪P■)=TUTTI I NUMERI PARI"
360 PRINT"◪IMPAIR■ (◪I■)=TUTTI I NUMERI DISPARI"
370 PRINT
380 PRINT"◪MANQUE■ (◪MA■)=I NUMBERI ◪1-18■"
390 PRINT"◪PASSE■ (◪PA■)=I NUMBERI ◪19-36■"
400 PRINT
410 PRINT"◪D1■=NUMBERI ◪1-12■"
420 PRINT"◪D2■=NUMBERI ◪13-24■"
430 PRINT"◪D3■=NUMBERI ◪25-36■"
440 PRINT
450 PRINT"◪R■=NUMERI ROSSI"
460 PRINT"◪B■=NUMERI NERI"
470 PRINT:PRINT"PREMI UN TASTO PER CONTINUARE"
480 GETA$:IFA$=""THEN480
490 PRINT"⊐"
500 GOSUB2210
510 FORZZ=1TO5
520 X$="*":Z$="⊐←":U$="⊐ "
530 X=INT(RND(1)*37)
540 IFX<10THENY$="⊐ ":GOTO560
```

```
550 Y$="⌝I"
560 GOSUB2180
570 IFG=1THENA$="R":GOTO590
580 A$="B"
590 G=0:PRINT"▒▒▒▒▒▒":IFX<10THENPRINTTAB(6)" "X"▒"A$:GOTO610
600 PRINTTAB(5)" "X"▒"A$
610 IFX>12THEN630
620 ONX+1GOTO660,690,700,710,720,730,740,750,760,770,780,790,800
630 IFX>25THEN650
640 ONX-12GOTO810,820,830,840,850,860,870,880,890,900,910,920,930
650 ONX-25GOTO940,950,960,970,980,990,1000,1010,1020,1030,1040
660 PRINT"▒▒":PRINTTAB(17)X$
670 GOSUB2000
680 PRINTTAB(17)Y$:GOTO1050
690 K=1:GOTO2480
700 K=2:GOTO2480
710 K=3:GOTO2480
720 K=1:GOTO2490
730 K=2:GOTO2490
740 K=3:GOTO2490
750 K=1:GOTO2500
760 K=2:GOTO2500
770 K=3:GOTO2500
780 K=1:GOTO2510
790 K=2:GOTO2510
800 K=3:GOTO2510
810 K=1:GOTO2520
820 K=2:GOTO2520
830 K=3:GOTO2520
840 K=1:GOTO2530
850 K=2:GOTO2530
860 K=3:GOTO2530
870 K=1:GOTO2540
880 K=2:GOTO2540
890 K=3:GOTO2540
900 K=1:GOTO2550
910 K=2:GOTO2550
920 K=3:GOTO2550
930 K=1:GOTO2560
940 K=2:GOTO2560
950 K=3:GOTO2560
960 K=1:GOTO2570
970 K=2:GOTO2570
980 K=3:GOTO2570
990 K=1:GOTO2580
1000 K=2:GOTO2580
1010 K=3:GOTO2580
1020 K=1:GOTO2590
1030 K=2:GOTO2590
1040 K=3:GOTO2590
1050 NEXTZZ
1060 GOSUB2000
1070 GOSUB2170
1080 PRINT"          E'IL TUO TURNO !"
1090 GOSUB2000:GOSUB2170
1100 INPUT"QUANTO DENARO ($) HAI ";A5
1110 GOSUB2170
1120 INPUT"▒NUMERO▒ OR ▒SERIE▒";B$
1130 GOSUB2170
1140 INPUT"QUANTO VUOI PUNTARE ";C
1150 IFC<=A5THEN1200
1160 GOSUB2170:PRINT"E' PIU' DI QUANTO HAI IN DEPOSITO !":GOSUB2000
1170 GOSUB2170:GOSUB2000
1180 GOSUB2170:GOTO1140
1190 GOSUB2130:GOSUB2170
1200 IFVAL(B$)>0ANDVAL(B$)<37THEN1790
1210 IFB$="P"THEN1400
1220 IFB$="I"THEN1450
1230 IFB$="MA"THEN1500
1240 IFB$="PA"THEN1530
1250 IFB$="D1"ORB$="D2"ORB$="D3"THEN1570
1260 IFB$="R"THEN1620
```

```
1270 IFB$="B"THEN1650
1280 IFB$>"T"ANDB$<"TN"ANDRIGHT$(B$,1)<>"J"THEN1310
1290 GOSUB2170:PRINT"NON VALE , PROVA DI NUOVO";:GOSUB2000
1300 GOSUB2170:GOTO1120
1310 GOSUB2080
1320 IF(B$="TA"ANDX<4)OR(B$="TB"ANDX>3ANDX<7)THEN1390
1330 IF(B$="TC"ANDX>6ANDX<10)OR(B$="TD"ANDX>9ANDX<13)THEN1390
1340 IF(B$="TE"ANDX>12ANDX<16)OR(B$="TF"ANDX>15ANDX<19)THEN1390
1350 IF(B$="TG"ANDX>18ANDX<22)OR(B$="TH"ANDX>21ANDX<25)THEN1390
1360 IF(B$="TI"ANDX>24ANDX<28)OR(B$="TK"ANDX>27ANDX<31)THEN1390
1370 IF(B$="TL"ANDX>30ANDX<34)OR(B$="TM"ANDX>33)THEN1390
1380 GOTO1720
1390 W=12:GOTO1680
1400 GOSUB2080
1410 J=1
1420 IFJ=XTHEN1720
1430 J=J+2:IFJ<36THEN1420
1440 GOTO1670
1450 GOSUB2080
1460 J=1
1470 IFJ=XTHEN1670
1480 J=J+2:IFJ<36THEN1470
1490 GOTO1720
1500 GOSUB2080
1510 IFX>0ANDX<19THEN1670
1520 GOTO1720
1530 GOSUB2080
1540 IFX>0ANDX<19THEN1720
1550 GOTO1670
1560 END
1570 GOSUB2080
1580 IFB$="D1"ANDX>0ANDX<13THENW=2:GOTO1680
1590 IFB$="D2"ANDX>12ANDX<25THENW=2:GOTO1680
1600 IFB$="D3"ANDX>24THENW=2:GOTO1680
1610 GOTO1720
1620 GOSUB2080:GOSUB2180
1630 IFG=1THENG=0:GOTO1670
1640 GOTO1720
1650 GOSUB2080
1660 G=0:GOSUB2180:IFG=1THENG=0:GOTO1720
1670 W=1
1680 PRINT"▓▓▓▓▓▓▓"TAB(6)X"▌ ":GOSUB2160
1690 PRINT"                    ▣HAI VINTO !▀        ⊤⊤⊤"TAB(24);STR$(A5+W*C);"
1700 GOSUB2000:PRINT"⊤⊤":GOSUB2170
1710 A5=A5+W*C:GOTO1120
1720 PRINT"▓▓▓▓▓▓▓"TAB(6)X"▌ "
1730 GOSUB2160
1740 PRINT"                    ▣HAI PERSO !▀        ⊤⊤⊤"TAB(24);STR$(A5-C);"      "
1750 A5=A5-C:GOSUB2000
1760 GOSUB2170
1770 IFA5>0THEN1120
1780 GOTO1830
1790 B=VAL(B$)
1800 GOSUB2080
1810 IFX=BTHENW=36:GOTO1680
1820 GOTO1720
1830 PRINT"⊤▣"
1840 PRINT"                 SEI"
1850 PRINT
1860 PRINT"           ▣A L  V E R D E ▀"
1870 PRINT:PRINT:PRINT
1880 PRINT"SE VUOI DESIDERI CONTINUARE , PRENDI ALTRO DENARO"
1890 PRINT"E PREMI UN TASTO PER CONTINUARE"
1900 GETA$:IFA$=""THEN1900
1970 GOSUB2210:GOTO1070
1980 END
1990 PRINTTAB(28)Z$
2000 FORI=1TO1000:NEXTI:RETURN
2010 PRINTTAB(28)U$:GOTO1050
2020 PRINTTAB(15)X$:RETURN
2030 PRINTTAB(15)Y$:GOTO2010
2040 PRINTTAB(18)X$:RETURN
```

```
2050 PRINTTAB(18)Y$:GOTO2010
2060 PRINTTAB(21)X$:RETURN
2070 PRINTTAB(21)Y$:GOTO2010
2080 X=INT(RND(1)*37)
2090 IFX<>0THENRETURN
2094 PRINT"▒▒▒▒▒▒▒"TAB(6)X"■ "
2096 GOSUB2160
2100 PRINT"              ▒IL BANCO VINCE !■       ▒▒▒"TAB(24);STR$(A5-C);"   "
2105 A5=A5-C:PRINT"▒▒"
2110 IFA5<=0THENGOSUB2000:GOTO1830
2120 GOSUB2000:GOSUB2170:GOTO1120
2130 FORJ=1TO5:PRINT"▒SEI BLOCCATO !■"
2140 FORI=1TO70:NEXTI
2150 PRINT"SEI BLOCCATO !":FORI=1TO70:NEXTI:NEXTJ:RETURN
2160 FORI=1TO15:PRINT:NEXTI
2170 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";:FORE=0TO117:PRINT" ";:NEXTE:PRINT" "
2175 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";:RETURN
2180 IFX=1ORX=3ORX=5ORX=7ORX=9ORX=12ORX=14ORX=16ORX=18ORX=19THENG=1:RETURN
2190 IFX=21ORX=23ORX=25ORX=27ORX=30ORX=32ORX=34ORX=36THENG=1
2200 RETURN
2210 PRINT"▒"
2220 PRINT"              ──────────────────"
2230 PRINT"              |P|    0     |M|"
2240 PRINT"       ───Q───|A|──────────|A|"
2250 PRINT"       |──|──||S| ▒1■| 2| ▒3■ |N|"
2260 PRINT"       |  |  ||S| 4| ▒5■| 6 |Q|"
2270 PRINT"       |──|──||E| ▒7■| 8| ▒9■ |U|"
2280 PRINT"       ───────| |10|11|▒12■ |E|"
2290 PRINT"              ──────────────────"
2300 PRINT"              |P|13|▒14■|15 |I|"
2310 PRINT"              |A|▒16■|17|▒18■ |M|"
2320 PRINT"              |I|▒19■|20|▒21■ |P|"
2330 PRINT"              |R|22|▒23■|24 |.|"
2340 PRINT"              ──────────────────"
2350 PRINT"              |N|▒25■|26|▒27■ |R|"
2360 PRINT"              |O|28|29|▒30■ |O| HAI IN"
2370 PRINT"              |I|31|▒32■|33 |U| DEPOSITO $"
2380 PRINT"              |R|▒34■|35|▒36■ |G|"
2390 PRINT"              ──────────────E─"
2400 PRINT"              | |1.|2.|3. | |"
2410 PRINT"              ──────────────────"
2420 PRINT
2430 RETURN
2440 ONKGOTO2450,2460,2470
2450 GOSUB2020:GOSUB1990:GOTO2030
2460 GOSUB2040:GOSUB1990:GOTO2050
2470 GOSUB2060:GOSUB1990:GOTO2070
2480 PRINT"▒▒▒▒":GOTO2600
2490 PRINT"▒▒▒▒▒":GOTO2600
2500 PRINT"▒▒▒▒▒▒":GOTO2600
2510 PRINT"▒▒▒▒▒▒▒":GOTO2440
2520 PRINT"▒▒▒▒▒▒▒▒▒":GOTO2440
2530 PRINT"▒▒▒▒▒▒▒▒▒▒":GOTO2440
2540 PRINT"▒▒▒▒▒▒▒▒▒▒▒":GOTO2440
2550 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒":GOTO2440
2560 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒":GOTO2440
2570 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒":GOTO2440
2580 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒":GOTO2440
2590 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒":GOTO2440
2600 ONKGOTO2610,2620,2630
2610 PRINTTAB(16)X$:GOSUB1990:PRINTTAB(16)Y$:GOTO2010
2620 PRINTTAB(19)X$:GOSUB1990:PRINTTAB(19)Y$:GOTO2010
2630 PRINTTAB(22)X$:GOSUB1990:PRINTTAB(22)Y$:GOTO2010

READY.
```

TOPI

È un gioco per due giocatori.

Ognuno controlla un topo: il primo giocatore usa i tasti A, W, D, X, il secondo usa i tasti 4, 8, 6, 2.

Il giocatore, che per primo urta i bordi, la traccia degli altri giocatori o la sua, perde.

```
10 REM 3. TOPI
20 PRINT"]                REGOLE":PRINT
22 PRINT"OGNI GIOCATORE PUO' MUOVERE IL SUO TOPO IN QUATTRO DIREZIONI."
23 PRINT"LE CHIAVI A,W,D,X CONTROLLANO IL TOPO    DI SINISTRA."
24 PRINT"LE CHIAVI 4,8,6,2 CONTROLLANO IL TOPO    DI DESTRA."
25 PRINT"I TOPI NON DEVONO ATTRAVERSARE IL BORDO,LA PROPRIA TRACCIA O QUELLA"
26 PRINT"DEGLI ALTRI."
27 PRINT"IL GIOCO CONSISTE DI 6 ROUNDS."
28 PRINT
30 PRINT:PRINT"PREMI QUALSIASI TASTO PER CONTINUARE"
32 GETA$:IFA$=""THEN32
50 PRINT"]"
65 GOTO1135
68 REM START
69 GOSUB4000:REM BORDER
70 GOSUB3000:REM KEY
80 IFL$="W"THENLY=LY-1:IFLY<0THENLY=0
90 IFL$="D"THENLX=LX+1:IFLX>39THENLX=39
100 IFL$="A"THENLX=LX-1:IFLX<0THENLX=0
110 IFL$="X"THENLY=LY+1:IFLY>24THENLY=24
200 PL=1024+LY*40+LX:CL=55296+LY*40+LX
205 IFPEEK(PL)<>32THEN1000
210 POKEPL,87:POKECL,8
220 GOSUB2000
270 GOSUB3000
280 IFR$="8"THENRY=RY-1:IFRY<0THENRY=0
290 IFR$="6"THENRX=RX+1:IFRX>39THENRX=39
300 IFR$="4"THENRX=RX-1:IFRX<0THENRX=0
310 IFR$="2"THENRY=RY+1:IFRY>24THENRY=24
400 PR=1024+RY*40+RX:CR=55296+RY*40+RX
405 IFPEEK(PR)<>32THEN1010
410 POKEPR,81:POKECR,13
420 GOSUB2000
500 GOTO70
1000 FORI=1TO1000:NEXTI:PRINT"]▮▮▮▮▮▮▮▮▮▮▮IL GIOCATORE DI SINISTRA E' BLOCCATO"
1005 ZR=ZR+1:GOTO1100
1010 ZL=ZL+1
1012 FORI=1TO1000:NEXTI:PRINT"]▮▮▮▮▮▮▮▮▮▮▮IL GIOCATORE DI DESTRA   E' BLOCCATO"
1100 FORI=1TO2000:NEXTI:PRINT"▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮PUNTEGGIO"
1120 PRINT"▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮";:PRINTZL;:PRINT":";:PRINTZR:FORI=1TO3000:NEXTI
1130 IFZL+ZR=6THEN1140
1135 FORI=1TO500:NEXTI:GOTO68
1140 PRINT"▮▮▮▮▮▮▮▮▮▮E' STATO UN BUON GIOCO !"
1150 FORI=1TO3000:NEXTI
1160 PRINT"▮PREMI QUALSIASI TASTO PER INIZIARE UN                  ALTRO GIOCO"
1170 GETA$:IFA$=""THEN1170
1180 ZL=0:ZR=0:GOTO68
2000 FORI=1TO50:NEXTI
2010 RETURN
```

```
3000 GETA$
3010 IFA$<>"W"THEN3100
3020 L$=A$:RETURN
3100 IFA$<>"A"THEN3200
3120 L$=A$:RETURN
3200 IFA$<>"X"THEN3300
3220 L$=A$:RETURN
3300 IFA$<>"D"THEN3400
3320 L$=A$:RETURN
3400 IFA$<>"2"THEN3500
3420 R$=A$:RETURN
3500 IFA$<>"4"THEN3600
3520 R$=A$:RETURN
3600 IFA$<>"8"THEN3700
3620 R$=A$:RETURN
3700 IFA$<>"6"THEN3800
3720 R$=A$:RETURN
3800 RETURN
4000 S=1024:C=55296
4040 PRINT"♥":V=0
4050 FORH=0TO39:GOSUB5000:NEXTH:H=39
4060 FORV=0TO23:GOSUB5000:NEXTV:V=23
4070 FORH=39TO0STEP-1:GOSUB5000:NEXTH:H=0
4080 FORV=23TO0STEP-1:GOSUB5000:NEXTV
4090 LX=1:LY=23
4100 RX=38:RY=23
4110 L$="W":R$="8"
4120 RETURN
5000 POKE(S+V*40+H),102:POKE(C+V*40+H),4
5010 RETURN

READY.
```

SUPER TRE

In questo gioco vedrete sullo schermo un reticolo di 9 quadrati, numerati da 3 a 11. Ognuno di questi contiene altri 9 quadrati più piccoli, pure numerati da 3 a 11 (al posto del numero 7 compare il numero dei quadrati grandi in campo inverso). Sotto il campo di gioco sono mostrati due dadi.

Scopo di questo gioco è di occupare tre quadrati piccoli disposti su una fila, su una colonna o in diagonale o di riuscire a possedere un totale di cinque quadrati piccoli qualsiasi e quindi di vincere il quadrato grande. Vince chi per primo occupa tre quadrati grandi in fila, colonna o diagonale o un totale di cinque quadrati grandi.

Il dado stabilisce quale quadrato piccolo potete occupare.

Per esempio, se il dado mostra un quattro e un sei, allora potete occupare qualsiasi dei quadrati piccoli numerati con 10 (eccetto quelli già occupati dal vostro avversario), o qualsiasi quadrato piccolo nel quadrato più grande denominato 10, ad eccezione del quadrato centrale. Per occupare un quadrato battete per primo il numero del quadrato grande, poi una virgola e poi il numero del quadrato piccolo all'interno del quadrato grande. Se il computer tira il dado in modo tale che nessun gioco risulta possibile, allora prova una seconda volta e, se necessario, una terza. Dopo il terzo tentativo il gioco passa all'avversario.

Numeri speciali:

Se otterrete un dodici potrete occupare qualsiasi posizione libera sul campo di gioco.

Se tirate un sette, qualsiasi posizione centrale libera; ovvero qualsiasi posizione nel quadrato grande denominato 7.

Se otterrete un due (due uno) potrete cancellare qualsiasi posizione occupata dall'avversario,, ad eccezione delle posizioni già completate nel quadrato grande.

```
5 REM 4. SUPER TRE
10 DIMT%(12,11):A=INT(RND(2)*2+1)
20 PRINT"▣┌──────────────────────────────────────┐";
30 PRINT"▌(3) (4) (5) |(3) (4) (5) |(3) (4) (5) ▚ ■";
40 PRINT"▌            |            |            ▚ ■";
50 PRINT"▌(6) (▚3■) (8) |(6) (▚4■) (8) |(6) (▚5■) (8) ▚ ■";
60 PRINT"▌            |            |            ▚ ■";
70 PRINT"▌(9)(10)(11) |(9)(10)(11) |(9)(10)(11) ▚ ■";
80 PRINT"▌──────────────────────────────────────▚ ■";
90 PRINT"▌(3) (4) (5) |(3) (4) (5) |(3) (4) (5) ▚ ■";
100 PRINT"▌            |            |            ▚ ■";
110 PRINT"▌(6) (▚6■) (8) |(6) (▚7■) (8) |(6) (▚8■) (8) ▚ ■";
```

```
120 PRINT"|              |              |              ";
130 PRINT"|(9)(10)(11) |(9)(10)(11) |(9)(10)(11)  ";
140 PRINT"|-----------------------------------------";
150 PRINT"|(3) (4) (5) |(3) (4) (5) |(3) (4) (5)  ";
160 PRINT"|              |              |              ";
170 PRINT"|(6) (9) (8) |(6)(10) (8) |(6)(11) (8)  ";
180 PRINT"|              |              |              ";
190 PRINT"|(9)(10)(11) |(9)(10)(11) |(9)(10)(11)  ";
200 PRINT"------------------------------------------";
210 PRINT"";
220 PRINT" ┌───┐  ┌───┐"
230 PRINT"|    |  |    |"
240 PRINT"|    |  |    |"
250 PRINT"|    |  |    |"
260 PRINT" └───┘  └───┘"
270 PRINT"";
280 X=INT(RND(1)*6+1):Y=INT(RND(1)*6+1):Z=Y+X:IFZ=2THEN1520
290 FORR=3TO11:IFT%(Z,R)=0THENM=0:GOTO340
300 NEXTR
310 FORR=3TO11:IFT%(R,Z)=0THENM=0:GOTO340
320 NEXTR:M=M+1:IFM=3THEN1210
330 GOTO210
340 ONXGOTO350,360,370,380,390,400
350 PRINT"";:GOTO410
360 PRINT"*";:GOTO410
370 PRINT"";:GOTO410
380 PRINT"*";:GOTO410
390 PRINT"*";:GOTO410
400 PRINT"***";
410 B=B+1:IFB=2THENB=0:GOTO430
420 ONYGOTO350,360,370,380,390,400
430 PRINTTAB(13)"GIOCATORE"A"E'IL TUO TURNO"
440 INPUT"";I,J
450 IFZ=2THEN1270
460 IFZ=7THEN490
470 IFZ=12THEN500
480 IFJ=7THEN1260
490 IFI<>ZANDJ<>ZTHEN1260
500 IFI<3ORJ<3THEN1260
510 IFI>11ORJ>11THEN1260
520 IFT%(I,J)<>0THEN1260
530 PRINT"";:ONI-2GOTO540,670,680,690,700,710,720,730,740
540 ONJ-2GOTO640,550,560,570,750,580,590,600,610
550 PRINTSPC(4);:GOTO640
560 PRINTSPC(8);:GOTO640
570 PRINT"";:GOTO640
580 PRINT"";SPC(8);:GOTO640
590 PRINT"";:GOTO640
600 PRINT"";SPC(3);:GOTO620
610 PRINT"";SPC(7);
620 IFA=1THENPRINT"":GOTO660
630 PRINT"":GOTO660
640 IFA=1THENPRINT"":GOTO660
650 PRINT""
660 T%(I,J)=A+4:GOTO790
670 PRINTSPC(13);:GOTO540
680 PRINTSPC(26);:GOTO540
690 PRINT"";:GOTO540
700 PRINT"";SPC(13);:GOTO540
710 PRINT"";SPC(26);:GOTO540
720 PRINT"";:GOTO540
730 PRINT"";SPC(13);:GOTO540
740 PRINT"";SPC(26);:GOTO540
750 PRINT"";:IFI>9ANDA=1THENPRINT"":GOTO660
760 IFI>9THENPRINT"":GOTO660
770 IFA=1THENPRINT"":GOTO660
780 PRINT"":GOTO660
790 FORR=3TO11:IFT%(I,R)=5THENL=L+1
800 NEXTR:IFL=5THENL=0:GOTO940
810 L=0:FORR=3TO11:IFT%(I,R)=6THENL=L+1
820 NEXTR:IFL=5THENL=0:GOTO940
830 L=0:C=3
840 FORR=CTOC+2:Q=Q+T%(I,R):NEXTR:GOSUB910
```

```
850 C=C+3:IFC=12THENC=2:GOTO870
860 GOTO840
870 C=C+1:IFC=6THEN890
880 FORR=CTOC+6STEP3:Q=Q+T%(I,R):NEXTR:GOSUB910:GOTO870
890 FORR=3TO11STEP4:Q=Q+T%(I,R):NEXTR:GOSUB910
900 FORR=5TO9STEP2:Q=Q+T%(I,R):NEXTR:GOSUB910:GOTO1210
910 IFQ=15ORQ=18THEN940
920 Q=0
930 RETURN
940 PRINT"▒▒▒";
950 ONI-2GOTO1040,960,970,980,990,1000,1010,1020,1030
960 PRINTSPC(13);:GOTO1040
970 PRINTSPC(26);:GOTO1040
980 PRINT"▒▒▒▒▒▒";:GOTO1040
990 PRINT"▒▒▒▒▒▒";SPC(13);:GOTO1040
1000 PRINT"▒▒▒▒▒▒";SPC(26);:GOTO1040
1010 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒";:GOTO1040
1020 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒";SPC(13);:GOTO1040
1030 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒";SPC(26);
1040 IFA=2THEN1070
1050 FORR=1TO5:FORV=1TO12:PRINT"▒";:NEXTV:PRINT"████████████▒";:NEXTR
1060 T%(I,2)=5:FORR=3TO11:T%(I,R)=A+4:NEXTR:GOTO1110
1070 FORR=1TO5:FORV=1TO12:PRINT"▒";:NEXTV:PRINT"████████████▒";:NEXTR
1080 T%(I,2)=6:FORR=3TO11:T%(I,R)=A+4:NEXTR:FORR=3TO11:IFT%(R,2)=6THENQ=Q+1
1090 NEXTR:IFQ=5THEN1690
1100 GOTO1130
1110 FORR=3TO11:IFT%(R,2)=5THENQ=Q+1
1120 NEXTR:IFQ=5THEN1690
1130 Q=3:V=0
1140 FORR=QTOQ+2:V=V+T%(R,2):NEXTR:GOSUB1240
1150 Q=Q+3:IFQ=12THENQ=2:GOTO1170
1160 GOTO1140
1170 Q=Q+1:IFQ=6THEN1190
1180 FORR=QTOQ+6STEP3:V=V+T%(R,2):NEXTR:GOSUB1240:GOTO1170
1190 FORR=3TO11STEP4:V=V+T%(R,2):NEXTR:GOSUB1240
1200 FORR=5TO9STEP2:V=V+T%(R,2):NEXTR:GOSUB1240
1210 IFZ=2THEN210
1220 A=A+1:IFA=3THENA=1
1230 PRINT"▒▒":GOTO210
1240 IFV=15ORV=18THEN1690
1250 V=0:RETURN
1260 PRINTSPC(13)"VIOLAZ DI REGOLE▒████████████████"
1265 FORT1=1TO1000:NEXTT1
1267 PRINTSPC(13)"                    ███████████████████▒▒▒▒▒":GOTO430
1270 IFI<3ORJ<3ORI>11ORJ>11THEN1260
1280 IFT%(I,2)<>0THEN1260
1290 IFT%(I,J)=0THEN1260
1300 IFA=1ANDT%(I,J)=5THEN1260
1310 IFA=2ANDT%(I,J)=6THEN1260
1320 PRINT"▒▒▒";
1330 ONI-2GOTO1340,1430,1440,1450,1460,1470,1480,1490,1500
1340 ONJ-2GOTO1350,1360,1370,1380,1590,1390,1400,1410,1420
1350 PRINT"3";:GOTO1510
1360 PRINTSPC(4)"4";:GOTO1510
1370 PRINTSPC(8)"5";:GOTO1510
1380 PRINT"▒▒6";:GOTO1510
1390 PRINT"▒▒";SPC(8)"8";:GOTO1510
1400 PRINT"▒▒▒▒9";:GOTO1510
1410 PRINT"▒▒▒▒██10";:GOTO1510
1420 PRINT"▒▒▒▒";SPC(7)"11";:GOTO1510
1430 PRINTSPC(13);:GOTO1340
1440 PRINTSPC(26);:GOTO1340
1450 PRINT"▒▒▒▒▒▒";:GOTO1340
1460 PRINT"▒▒▒▒▒▒";SPC(13);:GOTO1340
1470 PRINT"▒▒▒▒▒▒";SPC(26);:GOTO1340
1480 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒";:GOTO1340
1490 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒";SPC(13);:GOTO1340
1500 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒";SPC(26);:GOTO1340
1510 T%(I,J)=0:GOTO210
1520 IFA=1THEND=6:GOTO1540
1530 D=5
1540 FORR=3TO11:FORV=3TO11:IFT%(R,2)=5ORT%(R,2)=6THEN1570
1550 IFT%(R,V)=DTHENM=0:GOTO340
```

```
1560 NEXTV
1570 NEXTR:M=M+1:IFM=3THEN1210
1580 GOTO210
1590 PRINT"▓▓▓▌";:ONI-2GOTO1600,1610,1620,1630,1640,1650,1660,1670,1680
1600 PRINT" (▓3■)";:GOTO1510
1610 PRINT" (▓4■)";:GOTO1510
1620 PRINT" (▓5■)";:GOTO1510
1630 PRINT" (▓6■)";:GOTO1510
1640 PRINT" (▓7■)";:GOTO1510
1650 PRINT" (▓8■)";:GOTO1510
1660 PRINT" (▓9■)";:GOTO1510
1670 PRINT" (▓10■)";:GOTO1510
1680 PRINT" (▓11■)";:GOTO1510
1690 PRINT"▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓"
1700 PRINT"▓     VINCE IL GIOCATORE  "A"          ■"

READY.
```

LOTTO

Questo programma potrà esservi di aiuto quando giocherete alla lotteria.

Il programma seleziona 6 numeri su 49 possibili. Per passare ad una nuova estrazione premete il tasto N.

```
5 REM 5. GIOCO DEL LOTTO
10 T0$="|    |    |    |    |    |    |    |"
20 T1$="┌────┬────┬────┬────┬────┬────┬────┐"
30 T2$="| 01 |02  |03  |04  |05  |06  |07  |"
40 T3$="| 08 |09  |10  |11  |12  |13  |14  |"
50 T4$="| 15 |16  |17  |18  |19  |20  |21  |"
60 T5$="| 22 |23  |24  |25  |26  |27  |28  |"
70 T6$="| 29 |30  |31  |32  |33  |34  |35  |"
80 T7$="| 36 |37  |38  |39  |40  |41  |42  |"
90 T8$="| 43 |44  |45  |46  |47  |48  |49  |"
100 T9$="└────┴────┴────┴────┴────┴────┴────┘"
110 S$="         "
120 PRINT"♥"
130 PRINT"▓"S$+T1$
140 PRINTS$+T2$
150 PRINTS$+T0$
160 PRINTS$+T1$
170 PRINTS$+T3$
180 PRINTS$+T0$
190 PRINTS$+T1$
200 PRINTS$+T4$
210 PRINTS$+T0$
220 PRINTS$+T1$
230 PRINTS$+T5$
240 PRINTS$+T0$
250 PRINTS$+T1$
260 PRINTS$+T6$
270 PRINTS$+T0$
280 PRINTS$+T1$
290 PRINTS$+T7$
300 PRINTS$+T0$
310 PRINTS$+T1$
320 PRINTS$+T8$
330 PRINTS$+T9$
340 PRINT"▓  PER CONTINUARE PREMI 'N' (=NUOVO)"
350 PRINT"  PER FINIRE PREMI     'F' (=FINE)"
360 X(0)=99
370 FORZ=1TO6
380 W=0
390 X(Z)=INT(RND(1)*49)
400 FORY=0TOZ-1
410 IFX(Z)=X(Y)THENW=1
420 NEXTY
430 IFW=1THEN380
440 XV=120*INT(X(Z)/7)
450 XH=4*(X(Z)-7*INT(X(Z)/7))
460 XX=1030+XV+XH:XC=55302+XV+XH
470 POKEXX,160:POKEXC,10
480 POKEXX+1,160:POKEXC+1,10
490 POKEXX+2,160:POKEXC+2,10
500 POKEXX+3,160:POKEXC+3,10
```

```
510 POKEXX+40,160:POKEXC+40,10
520 POKEXX+43,160:POKEXC+43,10
530 POKEXX+80,160:POKEXC+80,10
540 POKEXX+81,160:POKEXC+81,10
550 POKEXX+82,160:POKEXC+82,10
560 POKEXX+83,160:POKEXC+83,10
580 D=PEEK(XX+41)
590 POKEXX+41,D+128
600 D=PEEK(XX+42)
610 POKEXX+42,D+128
620 NEXTZ
630 GETA$:IFA$=""THEN630
640 IFA$="N"THEN130
650 END

READY.
```

COLLISIONE

Questo è un gioco simile al biliardo.

Sullo schermo compare un tavolo da gioco con due palle che voi dovete colpire con una terza palla.

L'unica cosa che dovete definire è l'angolo (da 0 a 360), che individua la direzione in cui desiderate spedire la palla.

```
100 REM 6. COLLISIONE
110 REM UN GIOCO SIMILE AL BILIARDO
120 PRINT"□"
400 GOSUB1000
410 GOSUB9000
420 GOSUB1500
500 P=P+1:IFP>NPTHENP=1:I=I+1
510 IFI>NITHEN7500
520 GOSUB2000
530 GOSUB2500
540 GOSUB3000
550 GOSUB3500
560 GOSUB4000
570 GOSUB4500
700 GOSUB5000
710 GOSUB5500
720 IFNB>=3THENGOSUB6000:GOTO500
730 GOSUB6500
740 IFCU>=8THENGOSUB6000:GOTO500
750 GOTO700
1000 RX=0:DX=0:RY=0:DY=0:TX=0:TY=0:XR=0:YU=0:CU=0:IX=0:IY=0:X1=0
1010 Y1=0:X2=0:Y2=0:A=0:B=0:NB=0:I=0:S1=0:S2=0:P=2:DIMP$(2):RETURN
1500 GOSUB8800:GOSUB8400:PRINT"                    ▧COLLISIONE■":PRINT
1510 INPUT"QUANTI GIOCATORI (1/2)";NP
1520 IFNP<1ORNP>2THEN1510
1530 IFNP=1THENINPUT"COME TI CHIAMI ";P$(1):GOTO1560
1540 INPUT"NOME DEL GIOCATORE N.1";P$(1)
1550 INPUT"NOME DEL GIOCATORE N.2";P$(2)
1560 INPUT"QUANTI TIRI ";NI
1570 NX=39:NY=18:SR=.865
1590 K=RND(TI):XR=NX-2
1600 YU=NY-2:RETURN
2000 GOSUB8800
2010 GOSUB8400:PRINT"PREPARATI ";P$(P):PRINT:PRINT
2020 GOSUB8900:GOSUB8800
2030 GOSUB8400:PRINT:PRINT
2040 PRINT"PROVA";I;"DI";NI,"TIRA, ";"▧";P$(P);"■":PRINT
2050 PRINT"PUNTI     ";P$(1);":";S1;:IFNP=2THENPRINT"    ";P$(2);":";S2
2060 RETURN
2500 GOSUB8600
2510 A=X:FORK=1TONX:PRINT"▧";:GOSUB8100:NEXTK:A=L:B=U:C=X
2520 FORK=1TONY-1:GOSUB8300:NEXTK:GOSUB8600:A=U:B=X:C=L
2530 FORK=1TONY-1:GOSUB8300:NEXTK:A=X:FORK=1TONX-1:GOSUB8100
2540 NEXTK:PRINT"■";:RETURN
3000 X1=INT(RND(1)*(NX-3))+1:Y1=INT(RND(1)*(NY-3))+2
3010 X2=INT(RND(1)*(NX-3))+1:Y2=INT(RND(1)*(NY-3))+2
3020 K=ABS(X1-X2):Q=ABS(Y1-Y2)
3030 IFK<3ANDQ<3THEN3010
3040 XB=INT(RND(1)*(NX-4))+2:YB=INT(RND(1)*(NY-4))+2
3050 K=(X1-XB)↑2+(Y1-YB)↑2:IFK<4THEN3040
3060 K=(X2-XB)↑2+(Y2-YB)↑2:IFK<4THEN3040
3070 RETURN
```

```
3500 LX=X1:LY=Y1:GOSUB7000:A=OH:B=OH:GOSUB8200
3510 A=L:B=L:C=D:GOSUB8300:A=OH:B=OH:GOSUB8200
3520 LX=X2:LY=Y2:GOSUB7000:A=OH:B=OH:GOSUB8200
3530 A=L:B=L:C=D:GOSUB8300:A=OH:B=OH:GOSUB8200
3540 LX=XB:LY=YB:GOSUB7000:A=AS:GOSUB8100:RETURN
4000 GOSUB8400
4010 INPUT"ANGOLO (0-360) ";G:G=G*π/180
4020 DX=COS(G):DY=SIN(G):IFSR>1THENDX=DX/SR
4030 IFSR<1THENDY=DY*SR
4040 IFABS(DX)<.5ANDABS(DY)<.5THENDX=DX*2:DY=DY*2:GOTO4040
4050 RETURN
4500 LX=XB:LY=YB:GOSUB7000
4510 NB=0:CU=0:IX=XB:IY=YB:RX=XB+.49:RY=YB+.49
4520 GOSUB8900:A=SP:B=L:GOSUB8200
4530 RETURN
5000 RX=RX+DX:RY=RY+DY
5010 TX=INT(RX):TY=INT(RY)
5020 A=N:B=N:IFTX>IXTHENA=R:GOTO5040
5030 IFTX<IXTHENA=L
5040 IFTY>IYTHENB=U:GOTO5060
5050 IFTY<IYTHENB=D
5060 GOSUB8200:IX=TX:IY=TY
5070 A=AS:B=L:GOSUB8200
5080 A=SP:B=L:GOSUB8200
5090 RETURN
5500 IFNB>1THEN5540
5510 IFIX<X2ORIX>X2+1ORIY<Y2-1ORIY>Y2THEN5540
5520 A=PL:B=L
5530 GOSUB8200:NB=NB+2
5540 IFNB=1ORNB=3THEN5570
5550 IFIX<X1ORIX>X1+1ORIY<Y1-1ORIY>Y1THEN5570
5560 A=PL:B=L:GOSUB8200:NB=NB+1
5570 RETURN
6000 GOSUB8400
6010 PRINT
6020 IFNB<3THENPRINT"▓NIENTE PUNTI !■";:GOTO6070
6030 SC=4-ABS(CU-4):PRINT"QUESTO TIRO VALE :";SC:IFP=1THENS1=S1+SC
6040 IFP=2THENS2=S2+SC
6050 PRINT:PRINT:PRINT"SCORE     ";P$(1);":";S1
6060 IFNP=2THENPRINT"     ";P$(2);"-";S2
6070 FORQ=1TO3:GOSUB8900:NEXTQ:RETURN
6500 IFIX>1ANDIX<XRTHEN6530
6510 CU=CU+1
6520 A=BL:GOSUB8100:DX=DX*-1
6530 IFIY>1ANDIY<YUTHEN6550
6540 CU=CU+1:A=BL:GOSUB8100:DY=DY*-1
6550 RETURN
7000 GOSUB8600:A=R
7010 FORK=1TOLX:GOSUB8100:NEXTK
7020 A=U:FORK=1TOLY:GOSUB8100:NEXTK
7030 RETURN
7500 PRINT"▓E' FINITO !":END
8100 PRINTCHR$(A);
8110 RETURN
8200 PRINTCHR$(A);CHR$(B);
8210 RETURN
8300 PRINTCHR$(A);CHR$(B);CHR$(C);
8310 RETURN
8400 A=19
8410 GOSUB8100
8420 RETURN
8600 GOSUB8400
8610 A=17
8620 FORK=1TO24:GOSUB8100:NEXTK
8630 RETURN
8800 A=147
8810 GOSUB8100
8820 RETURN
8900 FORK=1TO1000:NEXTK
8910 RETURN
9000 R=29:L=157:U=145:D=17
9010 X=32:OH=215:AS=209:SP=32:BL=0:N=0:PL=43
9020 RETURN
```

FLIP-FLOP

In questo gioco cercate di prendere una pietra al vostro avversario e di farla diventare vostra. Attenzione non potete scambiarvi una pietra che è già stata scambiata dal vostro avversario.

Vince chi per primo forma una fila o una colonna completa.

```
5 REM 7. FLIP-FLOP
10 POKE59468,12:DIMA%(8,8),B%(8,8),RW(8),RS(8)
20 Z=215:T$(0)="▁▌▌":T$(1)=" ▌▌":T=80
30 FORI=1TO8:A$(I)=CHR$(I+64)
40 FORJ=1TO8
50 IF(I+J)AND1THENA%(I,J)=-1:GOTO70
60 A%(I,J)=0
70 NEXT:NEXT
80 GOSUB10000
90 PRINT"♥"TAB(9)"F L I P - F L O P▒▒
100 Y$="| | | | | | | | |":Z$="├─┼─┼─┼─┼─┼─┼─┼─┤"
110 PRINTTAB(10)"1 2 3 4 5 6 7 8"
120 PRINTTAB(9)"┌─┬─┬─┬─┬─┬─┬─┬─┐"
130 FORI=1TO7:PRINTTAB(8)A$(I)Y$:PRINTTAB(9)Z$:NEXT
140 PRINTTAB(8)A$(I)Y$:PRINTTAB(9)"└─┴─┴─┴─┴─┴─┴─┴─┘"
150 GOSUB500:IFT$="S"THEN240
170 S=INT(8*RND(TI)+1):W=INT(8*RND(TI)+1):IFA%(W,S)=0THEN170
180 GOTO200
190 GOTO6000
200 IFU=WANDV=STHEN190
210 PRINTTAB(25)"▒IO SCELGO : ";A$(W);RIGHT$(STR$(S),1):U=W:V=S
220 GOSUB600:GOSUB900:XT=XT+1:GOSUB500
230 IFSP=1ORCB=1THENGE=1:GOTO1100
240 GOSUB700
250 IFA%(W,S)=-1THENPRINT"TTT":GOTO240
260 IFU=WANDV=STHENPRINT"TTT":GOTO240
270 U=W:V=S
280 GOSUB600:GOSUB900:XT=XT+1:GOSUB500
290 IFSP=1ORCB=1THENGE=0:GOTO1000
300 GOTO190
499 REM *** PRINT-ROUTINE
500 PRINT"▒▒▒TIRO N. ▒▒▒▒▒"XT
510 FORI=1TO8:PRINTTAB(10);:W%(I)=0:S%(I)=0
520 FORJ=1TO8:PRINTCHR$(Z+6*A%(I,J))"▌▌";:B%(I,J)=A%(I,J):NEXT
530 PRINT:PRINT:NEXT:RETURN
599 REM ***  INVERS-ROUTINE
600 FORI=1TO8:A%(I,S)=NOT(A%(I,S))
610 A%(W,I)=NOT(A%(W,I)):NEXT
620 A%(W,S)=NOT(A%(W,S)):RETURN
699 REM ***  SCELTA DEL GIOCATORE
700 PRINT"▒TU SCEGLI ? ▌▌▌";
710 GOSUB800
720 IFT$<CHR$(65)ORT$>CHR$(72)THEN710
730 W=ASC(T$)-64:PRINTT$;
740 GOSUB800
750 IFT$=CHR$(20)THENPRINT"▌▌  ▌▌▌";:GOTO710
760 IFT$<CHR$(49)ORT$>CHR$(56)THEN740
770 PRINTT$;:S=VAL(T$):IFA%(W,S)=-1THENPRINT"TT":GOTO700
780 IFU=WANDV=STHENPRINT"TT":GOTO700
790 PRINTTAB(37)"  ":RETURN
799 REM *** INPUT (SCELTA DEL GIOCATORE)
800 OPEN1,0,0
810 GETT$:PRINTT$(ABS(X));:FORI=1TOT:NEXT:X=NOT(X):IFT$=""THEN810
820 A=ASC(T$):CLOSE1:RETURN
```

```
899 REM *** CALCOLO DEL PUNTEGGIO
900 FORI=1TO8:FORJ=1TO8
910 W%(I)=W%(I)+A%(I,J)
920 S%(I)=S%(I)+A%(J,I)
930 NEXT
940 IFW%(I)=-8ORS%(I)=-8THENSP=1
950 IFW%(I)=0ORS%(I)=0THENCB=1
960 NEXT:RETURN
1000 PRINT"■HAI VINTO !!! GIOCHI ANCORA ? (S/N)      ";
1010 GOTO1110
1100 PRINT"■HO VINTO !!! GIOCHI ANCORA ?  (S/N)      ";
1110 GETT$:IFT$=""THEN1110
1120 IFT$="S"THENCLR:PRINT"■■■■■  SI  ":GOTO10
1130 IFT$<>"N"THEN1110
1140 PRINT"■■■■■":END
6000 REM *** CALCOLO DELLE SCELTE
6010 FORI=1TO8:FORJ=1TO8
6040 ZW(I)=ZW(I)+B%(I,J)
6050 ZS(I)=ZS(I)+B%(J,I)
6060 NEXT:NEXT:WG=0:SG=0
6070 Z7=0:Z1=0:FORI=1TO8
6080 IFZW(I)=-7THENZ7=Z7+1
6090 IFZS(I)=-7THENZ7=Z7+1
6100 IFZW(I)=-1THENZ1=Z1+1:WG=I
6110 IFZS(I)=-1THENZ1=Z1+1:SG=I
6120 ZS(I)=0:ZW(I)=0
6130 NEXT:IFZ7>0ORZ1>0THEN6300
6140 W=INT(8*RND(TI)+1):S=INT(8*RND(TI)+1):IFB%(W,S)=0THEN6140
6150 FORI=1TO8:B%(I,S)=NOT(B%(I,S))
6160 B%(W,I)=NOT(B%(W,I)):NEXT
6170 B%(W,S)=NOT(B%(W,S))
6180 FORI=1TO8:FORJ=1TO8
6190 ZW(I)=ZW(I)+B%(I,J)
6200 ZS(I)=ZS(I)+B%(J,I)
6210 NEXT:NEXT
6220 Z7=0:Z1=0:FORI=1TO8
6230 IFZW(I)<=-7THENZ7=Z7+1
6240 IFZS(I)<=-7THENZ7=Z7+1
6250 IFZW(I)=-1THENZ1=Z1+1
6260 IFZS(I)=-1THENZ1=Z1+1
6270 ZS(I)=0:ZW(I)=0:NEXT
6281 FORI=1TO8:B%(I,S)=NOT(B%(I,S))
6282 B%(W,I)=NOT(B%(W,I)):NEXT
6283 B%(W,S)=NOT(B%(W,S))
6284 IFZ7<>0THEN6140
6290 GOTO200
6300 IFZ1=0THEN6140
6400 IFWG=0THEN6450
6410 FORI=1TO8:IFB%(WG,I)=-1THENSG=I
6415 NEXT
6420 FORI=1TO8:IFI=WGTHEN6440
6430 IFB%(I,SG)=-1THENW=I:S=SG
6440 NEXT:GOTO200
6450 FORI=1TO8:IFB%(I,SG)=-1THENWG=I
6455 NEXT
6460 FORI=1TO8:IFI=SGTHEN6480
6470 IFB%(WG,I)=-1THENW=WG:S=I
6480 NEXT:GOTO200
9999 REM *** TITOLI
10000 PRINT"■■■"TAB(6)"▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄"
10010 PRINTTAB(5)"■   F L I P  -  F L O P   ■"
10020 PRINTTAB(6)"▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀"
10030 PRINT"■■■UN GIOCO DI TATTICA CONTRO IL"
10040 PRINT"■COMPUTER"
10050 PRINT"■■■CONOSCI LE REGOLE ? (S/N)"
10051 GETT$:IFT$=""THEN10051
10052 IFT$="N"THEN30000
10053 IFT$<>"S"THEN10051
10055 PRINT"■■■TU HAI LE PIETRE BIANCHE (●)"
10060 PRINT"■■PREMI UN TASTO QUANDO SEI PRONTO"
10061 GETT$:IFT$=""THEN10061
10062 RETURN
29999 REM *** REGOLE
```

```
30000 PRINT"▯"TAB(12)"REGOLE"
30010 PRINT"▩▩OGNI GIOCATORE (UNO SEI TU, L'ALTRO"
30020 PRINT"E' IL COMPUTER) CERCA DI CAMBIARE LE"
30025 PRINT"PIETRE DELL'AVVERSARIO IN PIETRE DEL"
30030 PRINT"PROPRIO COLORE"
30040 PRINT"▩DURANTE IL CAMBIO TUTTE LE PIETRE"
30050 PRINT"NELLA FILA ORIZZONTALE E VERTICALE"
30060 PRINT"CORRISPONDENTI A QUELLA CAMBIATA"
30070 PRINT"SONO PURE CAMBIATE"
30080 PRINT"▩    (  ● —> o    E    o —> ●  )
30090 PRINTTAB(31)"▩▩▩▩______"
30100 PRINTTAB(30)"<▩RETURN▩>"
30110 GETT$:IFT$=""THEN30110
30120 PRINT"▯ESEMPIO :"
30130 PRINT"▩ECCO LA PIETRA CAMBIATA (INDICATA DALLA"
30140 PRINT"FRECCIA) E LE ATRE DELLE CORRISPONDENTI"
30142 PRINT"FILE ORIZZONTALE E VERTICALE▩▩"
30145 PRINTTAB(10)"4"TAB(25)"4"
30150 PRINTTAB(10)"●"TAB(25)"o"
30160 PRINTTAB(10)"●"TAB(25)"o"
30170 PRINTTAB(10)"o"TAB(25)"●"
30180 PRINTTAB(10)"o"TAB(25)"●"
30190 PRINTTAB(6)"Eo●●o●●o●"TAB(21)"E●oo●oo●o"
30200 PRINTTAB(9)"¬▩o"TAB(24)"¬▩●"
30210 PRINTTAB(8)"/ o"TAB(23)"/ ●"
30220 PRINTTAB(6)"E4  ●"TAB(21)"E4  o"
30230 PRINT
30240 PRINT
30250 PRINT
30260 PRINTTAB(31)"▩______"
30270 PRINTTAB(30)"<▩RETURN▩>"
30280 GETT$:IFT$=""THEN30280
30300 PRINT"▯▩▩UNA PIETRA CAMBIATA DALL'AVVERSARIO NON"
30310 PRINT"PUO' PIU' ESSERE CAMBIATA"
30320 PRINT
30330 PRINT
30340 PRINT"▩▩VINCE CHI RIESCE PER PRIMO A FORMARE"
30350 PRINT"▩UNA LINEA RETTA SIA IN ORIZZONTALE"
30360 PRINT"▩SIA IN VERTICALE"
30370 PRINT
30380 PRINT
30390 PRINT
30400 PRINTTAB(31)"▩▩______"
30410 PRINTTAB(30)"<▩RETURN▩>"
30420 GETT$:IFT$=""THEN30420
30430 GOTO10000

READY.
```

LABIRINTO

Scopo del gioco è di muoversi nel labirinto il più a lungo possibile senza subire una collisione.

I tasti di controllo sono 8, 2, 4, 6.

I tasti 8 e 2 o 4 e 6 non possono essere usati in successione (se provate vedrete che non funzioneranno).

```
5 REM 8. LABIRINTO
10 PRINT"□          L A B I R I N T O"
11 PRINT"         ================="
12 PRINT:PRINT"DEVI CERCARE DI MUOVERTI NEL LABIRINTO"
13 PRINT"IL PIU' A LUNGO POSSIBILE SENZA COLLISIONI":PRINT
14 PRINT"TASTO   8: SU"
15 PRINT"TASTO   2: GIU'"
16 PRINT"TASTO   4: SINISTRA"
17 PRINT"TASTO   6: DESTRA"
18 PRINT
20 PRINT"NON PUOI USARE I TASTI 8 , 2 , 4 , 6"
22 PRINT"IN SEQUENZA (NON LAVORERANNO)"
25 PRINT
30 PRINT"PREMI QUALUNQUE TASTO PER CONTINUARE"
35 GETA$:IFA$=""THEN35
40 B=1024:BC=55296
45 U$="2":O$="8":L$="4":R$="6"
50 G=100
100 PRINT"□"
110 FORY=1TO24:FORI=0TO7
120 X=INT((RND(1)+I)*5)
130 POKEB+40*Y+X,42:POKEBC+40*Y+X,13:NEXT:NEXT
150 FORI=OTO10:GETP$:NEXT
180 X=INT(RND(1)*38)
181 Y=INT(RND(1)*22)+1
182 Z=B+40*Y+X:IFPEEK(Z)<>32THEN180
183 POKEZ,81:POKEBC+40*Y+X,1
195 GETP$
196 IFP$<>U$ANDP$<>O$ANDP$<>L$ANDP$<>R$THEN195
197 T=TI:PA$="":GOTO210
200 GETP$
210 IFP$=U$ANDPA$<>O$THENPA$=P$:YI=1:XI=0
211 IFP$=R$ANDPA$<>L$THENPA$=P$:YI=0:XI=1
212 IFP$=L$ANDPA$<>R$THENPA$=P$:YI=0:XI=-1
213 IFP$=O$ANDPA$<>U$THENPA$=P$:YI=-1:XI=0
230 POKEZ,81:X=X+XI:Y=Y+YI
235 G=G*0.96:X0=TI+G
236 IFX0>TITHEN236
240 IFX<0ORX>39ORY<0ORY>24THEN300
250 Z=B+40*Y+X:ZC=BC+40*Y+X
260 IFPEEK(Z)=32THENPOKEZ,81:POKEZC,1:GOTO200
300 FORI=1TO8:X=TI
301 IFX+8>TITHEN301
303 IFX+9>TITHEN303
304 NEXT
320 PRINT"□IL TUO TEMPO E':";(TI-T-25)/60;"SECONDS"
322 X=INT(TI-T-25)/60
330 FORI=1TO10:GETA$:NEXT
350 PRINT"ANCORA  (S/N)"
355 GETSS$:IFSS$=""THEN355
360 P=PEEK(515):IFP<>22ANDPO=45THEN360
361 IFP=45THEN50
363 IFSS$="S"THEN100
370 PRINT"BYE BYE !"
380 FORI=1TO10:GETA$:NEXT

READY.
```

ATTACCO

In questo gioco potete scegliere di essere l'attaccante o il difensore di un castello.

L'attaccante si muove da un campo a quello vicino seguendo un cammino che viene poi ostruito e non può essere più ripetuto.

Il difensore può muoversi solo entro un campo limitato dalle lettere A, B e C.

L'attaccante cerca di entrare nel castello attraverso una posizione dove si trova il difensore e se ci riesce vince.

Entrambi i giocatori sono costretti a muoversi. Per finire il gioco premete il tasto S.

```
10 REM 9. ATTACCO
15 GOTO8000
20 FORI=1TO1700:NEXT
21 PRINTB$;C$:PRINTC$:PRINTC$:RETURN
30 GETA$:IFA$=""THEN30
31 RETURN
100 CLR
110 PRINT"▓UN MOMENTO  !"
120 DIMA(15,15),F(18)
130 Z1=209:Z2=170:Z3=102
140 B$="▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓"
141 C$="                                        "
145 W$="ABCBCCAABBCACBA"
150 FORI=1TO15
160 N=4:IFI=7ORI=8THENN=5
170 IFI=12ORI=15THENN=3
175 IFI=13ORI=14THENN=2
180 FORJ=1TON:READX:A(I,X)=1
190 NEXTJ:NEXTI
200 FORI=1TO18
220 READY:F(I)=Y
230 NEXTI
240 D1=PEEK(144):D2=PEEK(145)
300 GOSUB10000:X=1:X1=16
310 POKE1024+F(X),Z1:POKE1024+F(X1),Z2:POKE55296+F(X),13:POKE55296+F(X1),4
315 IFD=1THENSM=1:GOTO3000
320 PRINTB$;"▓ATTACCO▓ (A) O ▓DIFESA▓ (D)?":GOSUB30
330 IFA$="D"THENSM=1:GOTO3000
340 GOSUB21:GOTO1030
1000 GOSUB20
1030 PRINTB$;"ATTACCANTE SUL CAMPO";X;:INPUT"▓; VA A ";Y$:IFY$="S"THEN6000
1040 Y=VAL(Y$)
1041 IFX<12ANDY=0THENPRINT"▓JOLLY !":GOTO1000
1042 IFY=0THEN6000
1045 IFY>15THENPRINT"▓IL CAMPO NON ESISTE !":GOTO1000
1050 IFA(X,Y)=0THENPRINT"▓NON POSSIBILE !":GOTO1000
1060 IFA(X,Y)=2ORA(Y,X)=2THENPRINT"▓VIA BLOCCATA   !":GOTO1000
1070 GOSUB7000:IFSM=1THEN4000
2000 Y1=ASC(MID$(W$,X,1))-49:GOTO2060
2050 Y1=RND(-TI):Y1=INT(RND(1)*3+1)+15
2060 IFY1=X1THEN2050
2070 PRINTB$;"IL DIFENSORE VA A ";"▓ "CHR$(Y1+49)" ▓";
2075 GOSUB20
```

```
2080 POKE1024+F(X1),160:POKE1024+F(Y1),Z2
2085 POKE55296+F(X1),4:POKE55296+F(Y1),4
2090 X1=Y1
2095 IFSM=0THEN1030
3000 GOSUB21
3010 IFZ=0THENY=INT(RND(1)*4+2):GOTO1070
3030 M=0:FORI=1TO15:IFA(X,I)=1THENM=M+1:M(M)=I
3050 NEXT
3060 IFM=0THEN6000
3070 IFX<12THEN3210
3140 IFX1=16THENIFX=13ORX=14THENY=X1
3142 IFX1=17THENIFX=12ORX=13THENY=X1
3144 IFX1=18THENIFX=14ORX=15THENY=X1
3200 IFY>15THEN7000
3210 FORI=MTO1STEP-1
3215 IFPEEK(33019)=81ORPEEK(33031)=81THENY=M(1)
3220 IFASC(MID$(W$,M(I),1))-49=X1THENY=M(I):GOTO1070
3230 NEXT
3234 IFM<3THENY=M(1)
3240 Y=M(INT(RND(1)*M+1)):GOTO1070
4000 IFD=1THEN2000
4010 PRINTB$;"TOCCA A TE:"
4020 GOSUB30:Y1=ASC(A$)-49
4022 IFY1=34THEN6000
4025 IFY1<16ORY1>18THENPRINT"■SEI DISTRATTO            ":GOTO4040
4030 IFX1=Y1THENPRINT"     ■ DEVI ■ MUOVERE !":GOTO4040
4035 GOTO2080
4040 GOSUB20:GOTO4010
6000 GOSUB21:IFY$="S"THENA$=Y$
6004 IFA$="S"THENPRINTB$;"■NON AVEVI ALCUNA POSSIBILITA'   ":GOTO6200
6005 IFM=0ANDSM=1THENPRINTB$;"■PENSO SIA STATO UN ERRORE  !":GOTO6200
6010 X=ASC(Y$)-49
6020 IFX=X1THEN6100
6030 PRINTB$;"■HA-HA!!! H A I   P E R S O !":GOTO6200
6100 IFSM=1THENPRINTB$;"■HO VINTO , COME SEMPRE      !":GOTO6200
6110 PRINTB$;"■CONGRATULAZIONI■ HAI VINTO !"
6200 PRINT"UN ALTRO GIOCO ? (S/N)";:GOSUB30
6210 IFA$="S"THEN100
6500 END
7000 IFSM=0THEN7010
7005 PRINTB$;"L'ATTACCANTE VA DA ";X;" A  ";
7006 IFY<16THENPRINTY:GOTO7010
7008 PRINTCHR$(Y+49)
7010 H=X:FORI=1TO6
7020 IFI>3THENH=Y
7030 POKE1024+F(H),Z1:POKE55296+F(H),13
7040 FORJ=1TO200:NEXT
7050 POKE1024+F(H),Z3:POKE55296+F(H),13
7055 FORJ=1TO200:NEXT:NEXTI
7060 POKE1024+F(H),Z1:POKE55296+F(H),13
7065 IFY=X1THEN6100
7070 A(X,Y)=2:A(Y,X)=2
7071 POKE144,D1:POKE145,D2
7072 FORI=1TO28:READJ,H,B
7073 IFJ=XANDH=YTHENPOKE1024+B,81:POKE55296+B,13:GOTO7080
7074 IFJ=YANDH=XTHENPOKE1024+B,81:POKE55296+B,13:GOTO7080
7075 NEXT
7080 X=Y:Z=Z+1:PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■";Z
7090 GOSUB21
7100 RETURN
8000 PRINT"■"
8010 PRINT"VUOI VEDERE UNA PARTITA DIMOSTRATIVA ?  (S/N)"
8020 GETA$:IFA$=""THEN8020
8030 IFA$="S"THENCLR:D=1:GOTO110
8040 GOTO100
9000 DATA2,3,4,5,1,3,6,7,1,2,4,7,1,3,5,8
9010 DATA1,4,8,9,2,7,10,12
9020 DATA2,3,6,8,10,4,5,7,9,11
9030 DATA5,8,11,15,6,7,12,13,8,9,14,15
9040 DATA6,10,13,10,12
9050 DATA11,15,9,11,14
9100 DATA737,569,573,581,585,409,413,421,425,331
9105 DATA343,249,253,261,265,257,93,101
```

```
9200 DATA1,2,651,1,3,653,1,4,661,1,5,663
9212 DATA2,3,571,3,4,577,4,5,583,2,6,489,2,7,491
9214 DATA3,7,493,4,8,501,5,8,503,5,9,505
9216 DATA6,7,411,7,8,417,8,9,423
9218 DATA6,10,370,7,10,372,8,11,382,9,11,384,6,12,329,9,15,345
9220 DATA10,12,290,10,13,292,11,14,302,11,15,304
9222 DATA12,13,251,14,15,263
10000 PRINT"▒▒ATTACCO■──────────────────────▒CAMMINO■"
10011 PRINT"
10012 PRINT"                ▒ B      C ■
10013 PRINT"               /|◥▒     ■◤|\
10014 PRINT"              / | ◥▒   ■◤ | \
10015 PRINT"             /  |  ◥▒A■◤  |  \
10016 PRINT"            ▒───▒───▒ ■───▒───▒
10017 PRINT"          12|\ / 13   14 \ /|15
10018 PRINT"            | ▒10       11▒ |
10019 PRINT"            |/ \         / \|
10020 PRINT"            ▒───▒────────▒───▒
10021 PRINT"           6|  /|7      8|\  |9
10022 PRINT"            | / |        | \ |
10023 PRINT"            |/  |        |  \|
10024 PRINT"            ▒───▒────────▒───▒
10025 PRINT"           2 \  |3      4|  / 5
10026 PRINT"              \ |        | /
10027 PRINT"               \|        |/
10028 PRINT"                ▒▒▒▒▒▒▒▒▒▒
10029 PRINT"                    1
10030 RETURN

READY.
```

ARTIGLIERIA

In questo gioco due giocatori si sparano addosso attraverso una montagna.

I giocatori devono comunicare l'angolo e la forza di tiro e devono anche prendere in considerazione la velocità del vento.

```
5 REM 10. ARTIGLIERIA
10 DIMA(100),B(100),H(39),A1(2),B1(2),A2(2),B2(2),L(2)
15 PRINT"♥"
20 PRINT"▩ISTRUZIONI  ? (S/N)";:GET D$:IF D$="N"THEN41
21 IF D$<>"S"THEN 20
30 PRINT"♥","GIOCO DI ARTIGLIERIA"
31 PRINT:PRINT"DUE GIOCATORI SI SPARANO ADDOSSO"
32 PRINT"ATTRAVERSO UNA MONTAGNA."
33 PRINT:PRINT"A TURNO I GICATORI DEVONO INDICARE"
34 PRINT"L'ANGOLO DI TIRO (0-90)"
35 PRINT"E LA FORZA (5-10).":PRINT
36 PRINT"E DEVONO ANCHE CONSIDERARE IL VENTO."
37 PRINT:PRINT"PER AVERE IL DISEGNO DI UNA NUOVA        MONTAGNA RISPONDI 0."
39 PRINT:PRINT:PRINT"PREMI QUALSIASI TASTO PER COMINCIARE"
40 GET G$:IF G$="" THEN 40
41 PRINT"♥"
50 Z$="▩ ▜▌":FORX=1TO5:Z$=Z$+Z$:NEXTX
60 A$="▌":FORX=1TO6:A$=A$+A$:NEXTX:B$="▩":FORX=1TO5:B$=B$+B$:NEXTX
80 N=2
130 X1=INT(6+RND(1)*8)
140 X2=INT(25+RND(1)*8)
150 L(1)=INT(3+(X1-3)*RND(1))
160 L(2)=INT(X2+1+(36-X2)*RND(1))
170 H9=RND(1)*11+1
180 FOR I=1 TO X1:H(I)=H9:NEXTI
210 H9=RND(1)*11+1
220 FORI=X2TO39:H(I)=H9:NEXTI
250 IFABS(H(1)-H(39))<4THEN170
260 X3=X1+INT(RND(1)*(X2-X1-6))+3
270 H(X3)=RND(1)*11+9
290 D1=(H(X3)-H(1))/2
300 D2=(H(X3)-H(39))/2
310 IFD1*D2<=0THEN470
320 J=-180
330 K=180/(X3-X1+1)
340 H9=H(1)+D1
350 FORI=X1+1TOX3-1
360 J=J+K
370 H(I)=H9+COS(J*π/180)*D1
380 NEXTI
390 J=0
400 K=180/(X2-X3+1)
410 H9=H(39)+D2
420 FORI=X3+1TOX2-1
430 J=J+K
440 H(I)=H9+COS(J*π/180)*D2
450 NEXTI
460 GOTO550
470 D=(H(1)-H(39))/2
480 J=0
490 K=180/(X2-X1+1)
500 H9=H(39)+D
510 FORI=X1+1TOX2-1
```

```
520 J=J+K
530 H(I)=H9+COS(J*π/180)*D
540 NEXT I
550 PRINT"■■";
560 W=INT(100*(RND(1)-.5))
570 PRINTLEFT$(B$,22);
580 FORI=1TO39:H(I)=INT(H(I)+.5):NEXTI
590 FORX=1TO39:IFX=1THEN610
600 PRINT"■";
610 PRINTMID$(Z$,1,H(X)*4);
650 FORZ=1TOH(X):PRINT"■";:NEXTZ
660 NEXTX:PRINT"■"
670 PRINT"■";
680 PRINTMID$(B$,1,22-H(1));:IFL(1)=1THEN700
690 PRINTMID$(A$,1,L(1)-1);
700 PRINT"↑";
710 PRINT"■";
720 PRINTMID$(B$,1,22-H(39));
730 PRINTMID$(A$,1,L(2)-1);
740 PRINT"↑";:GET G$:GET G$
750 PRINT"■";
760 PRINTMID$(B$,1,24);
770 PRINT"            ";
780 K=1
790 IFW>=0THEN820
800 PRINT-W;" MPH VENTO VERSO SINISTRA";
810 GOTO840
820 PRINTW;" MPH VENTO VERSO DESTRA";
830 PRINT"■";
840 POKE54296,15:POKE54273,8:POKE54277,16:POKE54278,32+6
845 PRINT"■";:N=3-N:IFN=1THEN900
850 PRINTLEFT$(A$,28);"ANGOLO";
860 INPUT E:PRINT"■";
870 PRINTLEFT$(B$,1);LEFT$(A$,28);
875 PRINT "FORZA";
880 INPUTV:IFV>15THENV=15
890 GOTO910
900 PRINT"■";"
902 PRINT"
903 PRINT"ANGOLO";
905 INPUTE:PRINT"■■FORZA";:INPUTV:IFV>15THENV=15
910 POKE54276,128+1:POKE54276,128
920 K=K+1:IFE=0THEN130
930 A2(N)=A2(N)-2.86
940 E=90-E
950 IFN=1THEN970
960 E=-E
970 X1=L(N)
980 Y1=H(L(N))+1
990 T=0:M=0
1010 V1=SIN(E*π/180)*V
1020 V2=COS(E*π/180)*V*.7
1030 T=T+.5:M=M+1
1040 V1=V1+(V1+W*.5)/1500
1050 X=X1+(V1+W*.5/50)*T
1060 X=INT(X+.5)
1070 Y=Y1+(V2-T/3)*T
1080 Y=INT(Y+.5)
1090 Z=23-Y:PRINT"■";
1092 A(M)=Z:B(M)=X:IFX>39ORX<1THEN1350
1094 IFY-H(X)<=0THEN1340
1100 IFZ<0THEN1150
1110 IFZ=0THEN1125
1120 PRINTLEFT$(B$,Z);
1125 IFX=1THEN1140
1130 PRINTLEFT$(A$,X-1);
1140 PRINT"■"
1150 PRINT"■";
1160 IFM-4<1THEN1030
1162 IFA(M-4)=0THEN1180
1164 IFA(M-4)<0ORB(M-4)<=0THEN1030
1170 PRINTLEFT$(B$,A(M-4));
1180 PRINTLEFT$(A$,B(M-4)-1);
```

```
1190 PRINT" ";
1200 PRINT"■";
1220 GOTO1030
1230 IFM=1THEN1030
1232 IFB(M)<1ORB(M)>39THEN840
1234 IFABS(B(M-1)-L(N))<1THEN1250
1240 GOTO1260
1250 IFT>1THEN1510
1260 IFABS(B(M-1)-L(3-N))<1THEN1560
1270 IFT<1THEN1030
1280 GOTO1460
1290 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■ALTRO GIOCO (S/N)          ■■■■■■■■■";
1310 INPUT G$
1320 IFLEFT$(G$,1)<>"N"THEN15
1330 PRINT"■BYE! BYE!":PRINT
1335 END
1340 PRINT"■";
1350 FORJ=4TO1STEP-1
1360 IFM-J<=0THEN1440
1370 IFA(M-J)=0THEN1405
1380 IFA(M-J)<0ORB(M-J)<=0THEN1440
1400 PRINT LEFT$(B$,A(M-J));
1405 IFB(M-J)=1THEN1420
1410 PRINTLEFT$(A$,B(M-J)-1);
1420 PRINT" ";
1430 PRINT"■";
1440 NEXTJ:IFM=1THEN1600
1450 IFB(M)<1ORB(M)>39THEN1610
1455 GOTO1230
1460 IFB(M-1)=1THEN1470
1465 PRINTLEFT$(A$,B(M-1)-1);
1470 PRINTLEFT$(B$,23-H(B(M-1)));
1472 FORZ=1TO3:PRINT"■■■o/■■■■■?/";:FORX=1TO5:NEXTX
1474 PRINT"■■■■    ■■■■    ■■■■";:FORX=1TO5:NEXTX:NEXTZ:PRINT"■";
1480 GOTO 1600
1510 PRINT"■";
1520 PRINTLEFT$(A$,L(N)-1);
1530 PRINTLEFT$(B$,24-H(L(N)));
1540 FORZ=1TO10:PRINT"■■■■■■■■■■■■■■?■■■/■■■■■■■\?//■■■■";"■■";
1544 PRINT" ■■■■      ■■■■■■      ■■■■■■■■";
1545 FORX=1TO2:NEXTX:NEXTZ
1546 FORX=1TO50:NEXTX
1550 GOTO1290
1560 PRINT"■";
1570 PRINTLEFT$(A$,L(3-N)-1);
1580 PRINTLEFT$(B$,24-H(L(3-N)-1));
1590 GOTO1540
1600 REM
1610 W=W+INT(10*(RND(1)-.5))
1620 PRINT"■";LEFT$(B$,24);"              ";
1630 IFW>=0THENPRINT"   ■■■";W;" MPH VENTO VERSO DESTRA";:GOTO840
1640 PRINT"   ■■■";-W;" VENTO VERSO SINISTRA ";:GOTO840

READY.
```

QUADRATO MAGICO

In questo gioco dovete mettere dei numeri in un quadrato, in modo tale che la somma di ogni fila, colonna e diagonale dia per risultato 15.

```
100 REM 11. QUADRATO MAGICO
110 PRINT"□"
140 CLR:GOSUB600
150 PRINT:INPUT"INDICA POSIZIONE , NUMERO ";I,N
160 IFI<1ORI>9ORN<1ORN>9THEN180
170 IFA(I)=0ANDB(N)=0THEN190
180 PRINT:PRINT"INPUT ERRATO !...PROVA DI NUOVO":GOTO150
190 A(I)=N:B(N)=1:M=M+1
200 GOSUB600
210 GOSUB450
220 IFW=0THEN240
230 PRINT:PRINT"SPIACENTE , HAI PERSO !":GOTO420
240 IFM<5THEN260
250 PRINT"FFFFF":GOTO420
260 FORQ=1TO9
270 IFA(Q)>0THEN350
280 FORR=1TO9
290 IFB(R)>0THEN340
300 A(Q)=R
310 GOSUB450
320 IFW=0THEN370
330 Q1=Q:R1=R:W=0:A(Q)=0
340 NEXTR
350 NEXTQ
360 W=1:R=R1:Q=Q1:A(Q)=R
370 B(R)=1
380 PRINT:PRINT"IO METTO UN"R"IN POSITIONE"Q
390 GOSUB600
400 IFW=0THEN150
410 PRINT"BRAVO ! HAI VINTO"
420 INPUT"VUOI GIOCARE ANCORA (S/N)";Y$
430 IFY$="S"THEN140
440 END
450 FORX=1TO8
460 ONXGOTO470,480,490,500,510,520,530
470 J=1:K=2:L=3:GOTO540
480 K=4:L=7:GOTO540
490 K=5:L=9:GOTO540
500 J=4:L=6:GOTO540
510 J=3:L=7:GOTO540
520 K=6:L=9:GOTO540
530 J=7:K=8:GOTO540
540 IFA(J)=0ORA(K)=0ORA(L)=0THEN560
550 IFA(J)+A(K)+A(L)<>15THEN580
560 NEXTX
570 GOTO590
580 W=1
590 RETURN
600 PRINT:PRINTTAB(25)A(1);A(2);A(3)
610 PRINTTAB(25)A(4);A(5);A(6)
620 PRINTTAB(25)A(7);A(8);A(9)
630 PRINT
640 RETURN

READY.
```

GIROTONDO

In questo gioco vengono mostrati N numeri (N è definito nella linea 40 del programma) in una sequenza casuale.

Per vincere dovete disporre i numeri in ordine crescente, cambiando una certa parte dell'elenco.

Esempio:

l'elenco:

5 4 3 2 1 6 7

deve essere cambiato in:

1 2 3 4 5 6 7

nel modo seguente:
Quando il computer vi chiederà:
"QUANTI NE DEVO FAR GIRARE?" voi dovrete rispondere: 5.

```
10 REM 12. GIOSTRA
20 PRINT"J"
30 DIMA(20)
40 N=7:REM PUO' ESSERE CAMBIATO
50 A(1)=INT((N-1)*RND(1))+2
60 FORK=2TON
70 A(K)=INT(N*RND(1))+1
80 FORL=1TOK-1
90 IFA(K)=A(L)THEN70
100 NEXTL
110 NEXTK
120 PRINT
130 PRINT"LA LISTA DEI NUMERI E' LA SEGUENTE"
140 PRINT
150 T=0
160 GOSUB360
170 PRINT
180 INPUT"QUANTI NE DEVO FAR GIRARE ";R
190 IFR<=NTHEN220
200 PRINT:PRINT"IL MASSIMO E' ";N;" !!!"
210 GOTO170
220 T=T+1
230 FORK=1TOINT(R/2)
240 Z=A(K)
250 A(K)=A(R-K+1)
260 A(R-K+1)=Z
270 NEXTK
280 GOSUB360
290 FORK=1TON
300 IFA(K)<>KTHEN170
```

```
310 NEXTK
320 PRINT:PRINT:PRINT
330 PRINT"TI SERVONO"T"MOSSE !"
340 PRINT:PRINT:PRINT
350 END
360 PRINT
370 FORL=1TON
380 PRINTA(L);
390 NEXTL
400 PRINT
410 PRINT
420 RETURN

READY.
```

DISTANZA FRA DUE CITTÀ

Questo programma vi permette di calcolare la distanza tra due città qualsiasi della terra.

Quello che dovete conoscere sono le coordinate delle città (latitudine e longitudine).

Il programma vi mostra una mappa del mondo indicante i nomi lampeggianti delle città che avete selezionato, e contiene anche una lista di città con le loro coordinate.

```
1 REM 13. DISTANZA FRA DUE CITTA'
4 DIMW$(18)
5 PRINT"□"
10 PRINT"  CALCOLO DI DISTANZE"
15 PRINT
20 PRINT"  LE COORDINATE POSSONO ESSERE SCRITTE"
22 PRINT"  DALLA LINEA 2000 COME DATA"
23 PRINT"  FORMATO: DATA NOME,LONGITUDINE,          LATITUDINE"
25 FORI=1TO4000:NEXT
26 PRINT"□":GOSUB1100
27 Q=1024
50 PRINT:PRINT"■C■ALCOLO SECONDO LE COORDINATE"
51 PRINT"CITTA' DALLA ■L■ISTA  ■D■ISPLAY DELLA LISTA"
52 GETA$
53 IFA$="C"THEN99
54 IFA$="L"THEN1000
55 IFA$="D"THENLIST2000-
59 GOTO52
99 PRINT"□"
100 PRINT:PRINT"CALC. DI DISTANZA FRA DUE CITTA'"
102 PRINT"     (INPUT DELLE COORDINATE"
103 PRINT"     FORMATO:GRADI.MINUTI / OVEST E"
104 PRINT"     SUD NEGATIVI)"
110 PRINT"*"
111 PRINT"NOME DELLA PRIMA CITTA'";
113 INPUTN1$
115 INPUT"  LONGITUDINE ";L
120 GOSUB500
122 L1=G:L5=GR
125 INPUT"  LATITUDINE ";L
128 GOSUB500
130 B1=G:B5=GR
135 PRINT"NOME DELLA SECONDA CITTA'";
137 INPUTN2$
140 INPUT"  LONGITUDINE ";L
143 GOSUB500
145 L2=G:L6=GR
148 INPUT"  LATITUDINE ";L
150 GOSUB500
152 B2=G:B6=GR
160 V=SIN(B1)*SIN(B2)+COS(B1)*COS(B2)*COS(L2-L1)
161 IFV=0THENEM=π/2:GOTO166
162 EM=ATN(SQR(1-V*V)/V)
164 IFEM<0THENEM=EM+π
```

```
166 EM=EM*(180/π)*60
167 EK=EM*1.852
168 EM=INT(EM*1000+.5)/1000:EK=INT(EK*1000+.5)/1000
169 PRINT"♡":PRINT:PRINT:PRINT:PRINT
170 PRINT"LA DISTANZA DA ";N1$
175 PRINT"A ";N2$;"  E'"
178 PRINTEM;" MIGLIA, O";
180 PRINTEK;" KILOMETRI"
181 FORI=1TO3000:NEXT
182 B3=INT((90-B5)/10)+1
183 B4=INT((90-B6)/10)+1
190 G1=INT((L5+180)/10)+B3*40
192 G2=INT((L6+180)/10)+B4*40
200 PRINT"♡":GOSUB1150
202 EK=INT(EK)
210 PRINT:PRINT"DISTANZA DA ";N1$;" A ";N2$;" : ":PRINTEK;" KILOMETRI"
212 H1=PEEK(Q+G1):H2=PEEK(Q+G2)
213 H3=ASC(LEFT$(N1$,1))-64:H4=ASC(LEFT$(N2$,1))-64
214 IFG1=G2THEN224
215 FORI=1TO20
217 POKEQ+G1,H1:POKEQ+G2,H2
219 FORJ=1TO100:NEXT
220 POKEQ+G1,H3:POKEQ+G2,H4
222 FORJ=1TO500:NEXT
223 NEXT:GOTO230
224 FORI=1TO20
225 POKEQ+G1,H1:FORJ=1TO150:NEXT
226 POKEQ+G1,H3:FORJ=1TO500:NEXT
227 POKEQ+G2,H2:FORJ=1TO150:NEXT
228 POKEQ+G2,H4:FORJ=1TO500:NEXT
229 NEXT
230 PRINT"          ■PREMI <SPAZIO> PER CONTINUARE■"
240 GETA$:IFA$<>" "THEN240
250 GOTO50
500 Z=1
502 IFL<0THENZ=-1:L=ABS(L)
505 GR=(L-INT(L))*10/6+INT(L)
510 G=π/180*GR*Z
512 GR=GR*Z
515 RETURN
1000 PRINT"♡"
1010 RESTORE
1012 INPUT"NOME DELLA PRIMA CITTA' ";N$
1020 READB$,LA,BA
1022 IFB$="FINE"THEN1050
1024 IFB$=N$THEN1030
1025 B$=""
1029 GOTO1020
1030 PRINT:GOTO1060
1045 GOTO1036
1050 PRINT"LA CITTA' NON E' NELLA LISTA"
1055 GOTO50
1060 N1$=N$
1062 L=LA:GOSUB500
1064 L1=G:L5=GR
1065 L=BA:GOSUB500
1067 B1=G:B5=GR
1070 RESTORE
1075 INPUT"NOME DELLA SECONDA CITTA' ";N2$
1080 READB$,LA,BA
1083 IFB$="FINE"THEN1050
1085 IFB$=N2$THEN1087
1086 GOTO1080
1087 PRINT
1090 L=LA:GOSUB500
1091 L2=G:L6=GR
1092 L=BA:GOSUB500
1093 B2=G:B6=GR:GOTO160
```

```
1100 W$(1)="                                                  "
1102 W$(2)="                                                  "
1104 W$(3)="                                                  "
1106 W$(4)="                                                  "
1108 W$(5)="                                                  "
1110 W$(6)="                                                  "
1112 W$(7)="                                                  "
1114 W$(8)="                                                  "
1116 W$(9)="                                                  "
1118 W$(10)="                                                 "
1120 W$(11)="                                                 "
1122 W$(12)="                                                 "
1124 W$(13)="                                                 "
1126 W$(14)="                                                   "
1127 W$(15)="                    "
1128 W$(16)=" "
1130 W$(17)=" "
1132 W$(18)="                                                       "
1150 FORI=1TO18
1151 PRINTW$(I)
1154 NEXT
1195 RETURN
2000 DATA CHICAGO,-87.36,41.50
2002 DATA ST.LOUIS,-90.12,38.38
2006 DATA NEW ORLEANS,-90.04,29.56
2008 DATA DUESSELDORF,6.24,51.14
2010 DATA FRANCOFORTE,8.42,50.08
2014 DATA KAISERSLAUTERN,7.46,49.25
2016 DATA MONACO,11.34,48.09
2018 DATA BAD REICHENHALL,12.52,47.38
2020 DATA COLOMBO,80,7.08
2022 DATA SINGAPORE,103.5,1.25
2024 DATA JAKARTA,106.30,-6.17
2026 DATA CELEBES,120,-3.25
2028 DATA MILANO,9.07,45.27
2030 DATA BERLINO,13.2,52.3
2032 DATA BERNA,7.26,46.54
2034 DATA VIENNA,16.24,48.2
2036 DATA ROMA,12.2,41.55
2038 DATA CAIRO,30.15,30
2040 DATA MELBOURNE,145,-32.1
2042 DATA PECHINO,114.2,40
2044 DATA MOSCA,37.15,56.3
2046 DATA RIO DE JANEIRO,-43.5,-23
10000 DATA"FINE",0,0

READY.
```

CURVE

Il programma seguente vi permette di tracciare il grafico di funzioni matematiche. Per esempio:

FNA(X)=SIN(X)

oppure:

FNB(X)=X*X

```
5 REM 14. CURVE
10 PRINT"□";"@"
15 PRINT"QUESTO PROGRAMMA TRACCIA IL GRAFICO"
20 PRINT"DI DUE FUNZIONI MATEMATICHE."
26 PRINT:PRINT
30 PRINT"SE NELL'ANGOLO IN ALTO A DESTRA APPARE"
35 PRINT"IL SIMBOLO @ PUOI CONTINUARE"
40 PRINT"L'ESECUZIONE DEL PROGRAMMA PREMENDO"
42 PRINT"QUALSIASI TASTO."
43 GETM$:IFM$=""GOTO43
45 PRINT"□":PRINT"▓▓▓▓▓▓▓▓▓▓▓"
50 PRINT"▩";"SCRIVI ALLA LINEA 300: DEFFNA(X)=..."
60 PRINT"▩";"SCRIVI ALLA LINEA 310: DEFFNB(X)=..."
70 PRINT:PRINT:PRINT"300DEFFNA(X)=0":PRINT:PRINT"310DEFFNB(X)=0"
75 PRINT"RUN100"
80 PRINT"TTTTTTTTT":STOP
85 PRINT"NUOVO VALORE DI PARTENZA (S/N) "
90 GETL$:IFL$="S"THEN105
95 IFL$="N"THENEND
97 GOTO90
100 DIMU(39),V(39),W(39)
105 PRINT"□":INPUT"VALORE INIZIALE DI X ";A
110 INPUT "VALORE FINALE DI X ";B
120 INPUT"FATTORE DI SCALA ";C
130 INPUT"LIVELLO DELL'ASSE X ";D
140 E=38*(-A)/(B-A)+1
150 D=D+11
200 PRINT"□"
205 IFE<0ORE>38THEN220
210 FORF=1TO24:PRINTTAB(E);"|":NEXTF
220 PRINT"▩"
225 IFD<0ORD>24THEN300
230 FORG=1TOD:PRINT:NEXTG
240 PRINT" ────────────────────────────────────────"
300 DEFFNA(X)=X↑2
310 DEFFNB(X)=SIN(X)↑2+COS(X)↑2
320 H=(B-A)/38:L=0
330 FORX=ATOB STEPH
340 J=D-INT(FNB(X)*C+0.5):L=L+1
345 U(L)=INT(FNA(X)*1000000)/1000000
350 PRINT"▩";
355 IFJ>20ORJ<0GOTO375
360 FORK=0TOJ:PRINT:NEXTK
370 PRINTTAB(L);"o"
375 V(L)=INT(FNB(X)*1000000)/1000000:W(L)=INT(X*1000)/1000
380 M=D-INT(FNA(X)*C+0.5)
390 PRINT"▩";
```

```
395 IFM>20ORM<0GOTO420
400 FORN=0TOM:PRINT:NEXT
410 PRINTTAB(L);"*"
420 NEXTX
422 PRINT"▒";"@"
425 GETR$:IFR$=""GOTO425
427 PRINT"⊐":PRINT"▒▒▒▒▒▒▒"
430 PRINT"LISTA DEI VALORI ?"
450 GETM$:IFM$="S"GOTO480
460 IFM$="N"GOTO85
470 GOTO450
480 PRINT"⊐";"@"
490 PRINT" X:          FNA(X):    FNB(X):"
500 FORL=1TO19:PRINTW(L),U(L),V(L):NEXTL
510 GETM$:IFM$=""THEN510
520 PRINT"⊐";" X:          FNA(X):    FNB(X):"
530 FORL=20TO38:PRINTW(L),U(L),V(L):NEXTL
550 GOTO85

READY.
```

ZANZARE

Scopo del gioco è di ammazzare quante più zanzare possibili in tre minuti. Voi controllate lo schiacciazanzare con i tasti 2, 4, 6, 8.

```
10 REM 15. ZANZARE
59 REM=====REGOLE=====
60 PRINT"■"SPC(12)"*************"SPC(27)"*  ZANZARE   *"SPC(27)"*************
64 PRINT"SCOPO DEL GIOCO E' DI AMMAZZARE QUANTE"
66 PRINT"PIU' ZANZARE POSSIBILI IN UN TEMPO DI"
68 PRINT"180 SECONDI  CON UNO SCIACCIAZANZARE !"
69 PRINT
70 PRINT"LE CHIAVI '2','4','6','8' SERVONO PER"
72 PRINT"PER MANOVRARE LO SCHIACCIAZANZARE IN"
73 PRINT
74 PRINT"BASSO (2) SIN (4) DESTRA (6) ALTO (8)"
75 PRINT
76 PRINT"ALLA FINE VEDRETE QUANTE ZANZARE "
77 PRINT"AVRETE UCCISO."
78 PRINT"■           ■PREMI UN TASTO !■"
80 GETA$:IFA$=""THEN80
100 REM====ZANZARE====
170 DIMP(5),H(4):H(0)=0:DEAD=0
180 H(1)=40:H(2)=-1:H(3)=1:H(4)=-40
200 REM====BORDI====
210 PRINT"■":HO=1064:RU=HO+959:CO=55336:CU=CO+959
220 FORR=0TO39:POKEHO+R,211:POKECO+R,7
230 POKERU-R,211:POKECU-R,7:NEXT R
240 FORR=1TO23:POKEHO+40*R,211:POKECO+40*R,7
250 POKERU-40*R,211:POKECU-40*R,7:NEXT R
260 PRINT"■■■■■■■■■■■__TEMPO:__________
300 REM====POSIZIONE DI PARTENZA====
310 FOR K=1 TO 5:GOSUB 900
320 POKETP,123:P(K)=TP:POKETC,13:NEXT K
330 GOSUB900:POKETP,88:POKETC,13:PS=TP:PC=TC:T0=TI
399 REM====TEMPO====
400 ZEIT=INT((TI-T0)/60):PRINT"■■■■■■■■■■■■■■■■■■■■■"ZEIT"■":IFZEIT>=180THEN1000
420 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■";DEAD;"■"
499 REM==MOTO DELLO SCHIACCIAZANZARE==
500 GETZ$:IFZ$=""THEN550
510 IFZ$<>"2"ANDZ$<>"4"ANDZ$<>"6"ANDZ$<>"8"THENZ=0:GOTO550
520 Z=VAL(Z$):Z=Z/2
550 TS=PS+H(Z):PE=PEEK(TS):IFPE<>42GOTO580
560 FORK=1TO5:IFP(K)=TSTHENGOSUB800
570 NEXTK
580 IFPE<>32THEN600
590 POKEPS,32:PS=TS:POKEPS,88:POKEPS+54272,13
599 REM====MOTO DEI VOLI====
600 FORK=1TO5:Q=INT(5*RND(1)):TF=P(K)+H(Q):PE=PEEK(TF):IFPE=88THENGOSUB800:GOTO4
00
610 IFPE<>32THEN630
620 POKETF,123:POKETF+54272,13:POKETF-H(Q),32:P(K)=TF
630 NEXTK:GOTO400
799 REM===COLLISIONI===
800 Q=INT(5*RND(1)):TS=PS+H(Q):IFPEEK(TS)<>32THEN800
820 POKEPS,42:POKEPS+54272,13:PS=TS:POKEPS,88:POKEPS+54272,13:POKEP(K),32
825 GOSUB900:P(K)=TP:DEAD=DEAD+1
830 POKETP,123:POKETP+54272,13:RETURN
899 REM===POSIZIONI CASUALI===
900 R=INT(959*RND(1)):TP=HO+R:TC=CO+R:IFPEEK(TP)<>32THEN900
910 RETURN
```

```
1000 REM===TEMPO SCADUTO===
1010 PRINT"
1020 PRINT"
1021 PRINT"
1022 PRINT"
1023 PRINT"
1024 PRINT"
1025 PRINT
1026 PRINT"IN 180 SEC."DEAD" ZANZARE UCCISE !"
1028 REM===COMMENTI AI RISULTATI===
1030 ONINT(DEAD/4)+1GOSUB1100,1200,1300,1400,1500,1600,1700,1800,1900,2000,2100
1040 PRINT"                                         "SPC(4)""A$
1050 PRINT"                                         "SPC(4)""Z$
1055 IFK=9THENPRINT"    SPIACCICATORE DI ZANZARE !!       "
1060 FORI=1TO3000:NEXTI
1070 PRINT:PRINT
1080 INPUT"  VUOI GIOCARE ANCORA (SI/NO) ";ZZ$
1085 IFZZ$="SI"THEN1095
1090 IFZZ$="NO"THENPRINT"":END
1095 CLR:GOTO100
1096 REM
1097 REM===COMMENTI ALLE SUBROUTINE===
1098 REM===CHIAMATE DA ON..GOSUB===
1099 REM
1100 A$="   UNA PRESTAZIONE UN PO' DEBOLE !   "
1110 Z$="":RETURN
1200 A$=" PER 3 MINUTI  "+STR$(DEAD)+"   ZANZARE NON
1210 Z$="          SONO MOLTE                 ":RETURN
1300 A$="   IN AFRICA NON DORMIRESTI PER      "
1310 Z$="          5 ORE OGNI NOTTE  !        ":RETURN
1400 A$="          SODDISFACENTE  !           "
1410 Z$="   CONSIDERATI UN PRINCIPIANTE       ":RETURN
1500 A$="        NON TROPPO MALE  !           "
1510 Z$="        UNA BUONA PARTITA            ":RETURN
1600 A$="        M O L T O   B E N E !!!      "
1610 Z$="   SEI QUASI UN PROFESSIONISTA       ":RETURN
1700 A$="  VAI SEMPRE PIU'   F O R T E        "
1710 Z$="                                     ":RETURN
1800 A$="    FORTISSIMO ! SUPERLATIVO !       "
1810 Z$="    SEI IL PIU' GRANDE               ":K=9:RETURN
1900 FORK=1TO3000:NEXT:A$="      ♠♠♠♠♠♠♠♠♠♠♠♠♠♠♠♠♠♠♠♠♠♠♠♠♠
1910 Z$="SONO STRABILIATO PER LA TUA BRAVURA ":RETURN
2000 A$="  ECCELSO                            "
2010 Z$="                                     ":RETURN
2100 A$="    SEI IN ASSOLUTO IL PIU' GRANDE   "
2110 Z$="    CACCIATORE MONDIALE DI ZANZARE   ":RETURN

READY.
```

IL TEMPO NEL MONDO

Questo programma vi mostra il tempo centrale europeo (MEZ) e il tempo in altri 20 paesi.

```
5 REM 16. IL TEMPO NEL MONDO
7 PRINT"■":DC=54272
8 GOSUB4000
9 PRINTTAB(8)"■IL TEMPO NEL MONDO■":PRINT:PRINT:PRINT:PRINT
10 INPUT"CHE ORE SONO (6 CIFRE)         ";TI$
11 INPUT"GIORNO DELLA SETTIMANA (LU=1) ";X1
12 INPUT"GIORNO,MESE,ANNO (GG,MM,AAAA) ";X2,X3,X9:Y1=X3
13 DIMWO$(7),MO(12),MO$(12)
14 FORI=1TO7:READWO$(I):NEXT
15 FORI=1TO12:READMO(I),MO$(I):NEXT
16 DATALU,MA,ME,GI,VE,SA,"■DO■",31,GEN,28,FEB,31,MAR,30,APR,31,MAG,30,GIU
17 DATA31,LUG,31,AGO,30,SET,31,OTT,30,NOV,31,DIC
19 PRINT"■"
20 DATA151,143,146,140,132,148,137,141,133,160,160,143,142,160,160,148,136,133
21 DATA160,160,131,182,180
22 DATA1,1,2,1,1,6,1,1,1,1,1,20,1,1,1,1,1,2,1,3,1,1,1,1,1,1,1,1,1,20,1,1,1,1
23 DATA1,1,3,2,1,1,1,1,1,1,1,22,1,1,1,1,4,1,1,1,1,1,1,1,1,1,20,1,1,1,1
24 DATA8,1,1,1,1,1,1,1,2,20,1,1,1,4,1,1,1,1,1,2,1,2,2,20,1,1,1,5,1,1,1,1,1,2
25 DATA27,1,5,1,1,1,1,1,7,22,1,1,1,5,1,1,6,1,2,1,20,1,1,1,6,1,8,23,1,1,6,1,2,4
26 DATA1,1,1,22,1,1,6,2,5,1,1,1,22,1,15,1,2,22,17,64,4,1,1,1,1,1,1,1,1,25
27 DATA1,1,1,1,1,1,1,1,1,1,1,1,1,25,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0
28 DATA160,160,141,133,154,160,160,160,160,160,160,160,160,160,160,160,160,160
29 DATA160,160,160,160,160
95 REM
96 REM BORDI
97 REM
100 POKE1024,79:POKE55296,4
105 FORP=1025TO1062:POKEP,99:POKEP+DC,4:NEXT
110 POKE1063,80:POKE1063+DC,4
115 FORP=1103TO1983STEP40:POKEP,103:POKEP+DC,4:NEXT
120 POKE2023,122:POKE2023+DC,4
125 FORP=2022TO1985STEP-1:POKEP,100:POKEP+DC,4:NEXT
130 POKE1984,76:POKE1984+DC,4
135 FORP=1944TO1064STEP-40:POKEP,101:POKEP+DC,4:NEXT
136 POKE1066,45:POKE1101,43:POKE1066+DC,4:POKE1101+DC,4
140 FORP=1072TO1094:READA:POKEP,A:POKEP+DC,4:NEXT
141 REM
142 REM MAPPA DEL MONDO
143 REM
144 P=1151
145 READL:IFL=0THEN155
150 P=P+L:POKEP,42:POKEP+DC,13:GOTO145
151 REM
152 REM  OROLOGIO
153 REM
155 FORP=1294TO1301:READL:POKEP,L:POKEP+DC,7:NEXT
160 FORP=1341TO1421STEP40:READL:POKEP,L:POKEP+DC,7:NEXT
165 FORP=1461TO1454STEP-1:READL:POKEP,L:POKEP+DC,7:NEXT
170 FORP=1414TO1294STEP-40:READL:POKEP,L:POKEP+DC,7:NEXT
171 REM
173 REM
175 R=1240:S=81:U=R:V=42
178 PRINT"■■■■■■■■■■■■■■"TAB(30)WO$(X1)"         ""■■■■■■■■■■■■■"X2"■  "MO$(X3)" "
179 PRINT"■■■■■■■■■■■■■■■■■■■■"TAB(30)X9
180 REM
181 PRINT"■■■■■■■■■■"TAB(31)TI$
182 FORI=1TO5:POKER,S:POKEJ,K:FORH=1TO10:NEXTH:POKEU,V:POKEC,N
```

```
183 POKER+DC,13:POKEJ+DC,13:POKEU+DC,13:POKEC+DC,13:NEXTI
188 IFTI$="000000"THENGOTO3000
189 REMIFTI$="000001"THENTI$="235959"
190 XM=XM+1:IFXM=30THENQ=Q+1:XM=0:GOTO290
199 PRINT"SQQQQQQQQQQQQQQQQQQQQQQ"TAB(2)XX$
200 GOTO181
290 IFQ>10THENQX=Q-10
300 ONQGOSUB600,610,620,630,640,650,660,670,680,690
305 ONQXGOSUB700,710,720,730,740,750,760,770,780,790
306 IFQ=20THENQ=0:QX=0
310 GOTO181
350 END
400 REM
410 REM LISTA DELLE NAZIONI
420 REM
600 L$="EGITTO                ":T=1:GOSUB1000:J=1359:K=81:C=J:N=42:RETURN
610 L$="AMERICA COSTA EST     ":T=6:GOSUB2000:J=1352:C=J:RETURN
620 L$="AMERICA COSTA OVEST   ":T=9:GOSUB2000:J=1349:C=J:RETURN
630 L$="GIAPPONE              ":T=8:GOSUB1000:J=1369:C=J:RETURN
640 L$="USSR OVEST DI 172*     ":T=11:GOSUB1000:J=1209:C=J:RETURN
650 L$="USSR OVEST DI 112*     ":T=7:GOSUB1000:J=1243:C=J:RETURN
660 L$="ISOLE CANARIE         ":T=2:GOSUB2000:J=1396:C=J:RETURN
670 L$="CINA                  ":T=7:GOSUB1000:J=1326:C=J:RETURN
680 L$="MADAGASCAR            ":T=2:GOSUB1000:J=1602:C=J:RETURN
690 L$="AUSTRALIA             ":T=7:GOSUB1000:J=1609:C=J:RETURN
700 L$="PERU'                 ":T=6:GOSUB2000:J=1511:C=J:RETURN
710 L$="INDIA                 ":T=4:GOSUB1000:J=1404:C=J:RETURN
720 L$="GUATEMALA             ":T=7:GOSUB2000:J=1431:C=J:RETURN
730 L$="CANADA                ":T=6:GOSUB2000:J=1231:C=J:RETURN
740 L$="BRASILE CENTRO        ":T=5:GOSUB2000:J=1594:C=J:RETURN
750 L$="ALASKA                ":T=6:GOSUB2000:J=1188:C=J:RETURN
760 L$="MALESIA               ":T=6:GOSUB1000:J=1487:C=J:RETURN
770 L$="SOMALIA               ":T=2:GOSUB1000:J=1362:C=J:RETURN
780 L$="MEXICO OVEST..EST+1   ":T=8:GOSUB2000:J=1431:C=J:RETURN
790 L$="COREA                 ":T=8:GOSUB1000:J=1327:C=J:RETURN
999 REM
1000 REM   +OPERAZIONI
1001 REM
1005 Y$=MID$(TI$,3,2)
1010 X=VAL(LEFT$(TI$,2))+T:R$=" H"
1015 IFX>24THENX=X-24:R$=" M"
1020 IFVAL(LEFT$(TI$,2))>22THENX=0:R$=" M"
1025 IFX=24THENX=0:R$="M"
1030 XX$=L$+STR$(X)+" "+Y$+R$+"   "
1050 RETURN
1090 END
1999 REM
2000 REM   -OPERAZIONI
2001 REM
2010 D=VAL(LEFT$(TI$,2))
2020 X=D-T:R$=" H"
2030 IFD=0THENX=24-T:R$=" G"
2040 IFD>0ANDD<T+1THENX=24+D-T:R$=" G"
2044 IFX=24THENX=0:R$="H"
2045 XX$=L$+STR$(X)+" "+Y$+R$+"   "
2050 RETURN
2090 END
2900 REM
2950 REM CALCOLO E STAMPA DELLA DATA
2990 REM
3000 X1=X1+1:IFX1>7THENX1=1
3010 X2=X2+1
3020 IFX2>MO(Y1)THENX2=1:Y1=Y1+1:X3=X3+1:IFY1>12THENY1=1:X3=1:X9=X9+1
3030 REM
3040 REM CALCOLO DELL'ANNO BISESTILE
3050 REM
3060 IFX9/4=INT(X9/4)ORX9/400=INT(X9/400)THENMO(2)=29
3070 IFX9/4<>INT(X9/4)ORX9/400<>INT(X9/400)THENMO(2)=28
3100 FORI=1TO800:NEXT:GOTO178
4000 REM
4010 PRINTTAB(9)"IL TEMPO NEL MONDO":PRINT:PRINT:PRINT
4020 PRINT"QUESTO PROGRAMMA MOSTRA IL TEMPO MEDIO"
4030 PRINT"EUROPEO E L'ORA DI VARIE NAZIONI."
```

```
4035 PRINT
4040 PRINT"VENGONO ANCHE MOSTRATI : LA DATA, IL "
4050 PRINT"GIORNO DELLA SETTIMANA E L'ANNO."
4055 PRINT
4060 PRINT"GLI ANNI BISESTILI VENGONO CALCOLATI"
4065 PRINT"CON FEBBRAIO DI 29 GIORNI."
4070 PRINT
4080 PRINT"IN BASSO SULLO SCHERMO VIENE INDICATA"
4090 PRINT"L'ORA DI 20 NAZIONI."
4095 PRINT:PRINT"LA LINEA 189 SENZA REM FA SCORRERE"
4096 PRINT"I GIORNI PIU' IN FRETTA."
4100 PRINT:PRINT:PRINT"PREMI <■SPAZIO■> PER INIZIARE"
4200 GETW$:IFW$<>" "THEN4200
4250 PRINT"■"
4300 RETURN

READY.
```

CONCENTRAZIONE

Un gioco di carte che mette alla prova la vostra abilità di concentrazione.

Dovete estrarre delle carte da un mazzo senza prenderne di doppie. Se scegliete due volte la stessa carta avete perso.

```
10 REM 17. CONCENTRAZIONE
20 PRINT"♥"
30 PRINT"IMMAGINA DI AVERE UN PACCO DI CARTE"
40 PRINT"▒SCEGLI UNA CARTA :"
50 PRINT"SIA DI UNA CERTO SEME"
60 PRINT"SIA DI UN CERTO VALORE"
70 PRINT
80 PRINT"▒SE SCEGLI UNA CARTA NUOVA GUADAGNI PUNTI"
90 PRINT"▒SE SCEGLI UNA CARTA GIA' ESTRATTA"
100 PRINT"PERDI PUNTI"
110 PRINT
120 PRINT
130 PRINT
140 DIMP(4,13),N$(13),S$(4),T(4),S(13),M$(13),T$(4)
150 FORI=1TO4
160 FORJ=1TO13
170 P(I,J)=0
180 NEXTJ
190 NEXTI
200 FORI=1TO13
210 READN$(I),M$(I)
220 S(I)=0
230 NEXTI
240 FORI=1TO4
250 READS$(I),T$(I)
260 T(I)=0
270 NEXTI
280 DATAASSO,ASSO,DUE,DUE,TRE,TRE,QUATTRO
290 DATAQUATTRO,CINQUE,CINQUE,SEI,SEI,SETTE,SETTE
300 DATAOTTO,OTTO,NOVE,NOVE,DIECI,DIECI,FANTE
310 DATAFANTE,REGINA,REGINA,RE,RE
320 DATAFIORI,FIORI,QUADRI,QUADRI
330 DATACUORI,CUORI,PICCHE,PICCHE
340 FORZ=1TO52
350 X=INT(103*RND(1)+1)
360 IFX<52THEN600
370 X=X-4*INT(X/4)+1
380 IFT(X)>12THEN370
390 PRINT"ESTRAI UN ";S$(X);TAB(25);
400 INPUTR$
410 FORI=1TO13
420 IFR$=N$(I)THEN480
430 IFR$=M$(I)THEN480
440 NEXTI
450 PRINT"ERRORE ! PROVA DI NUOVO"
460 PRINT
470 GOTO390
480 IFP(X,I)=1THEN550
490 P(X,I)=1
500 K=K+T(X)+1
510 S(I)=S(I)+1
520 T(X)=T(X)+1
530 PRINT"O.K. IL PUNTEGGIO E' ";K
540 GOTO830
550 K=K-(20-T(X))
560 PRINT
```

```
570 PRINT"GIA' USATA ! RIPROVA     "
580 PRINT
590 GOTO390
600 X=X+1-13*(INT(X/13))
610 IFS(X)>3THEN600
620 PRINT"SCEGLI UN SEME DI ";M$(X);TAB(25);
630 INPUTR$
640 FORI=1TO4
650 IFR$=S$(I)THEN710
660 IFR$=T$(I)THEN710
670 NEXTI
680 PRINT"ERRORE ! PROVA DI NUOVO"
690 PRINT
700 GOTO620
710 IFP(I,X)=1THEN780
720 P(I,X)=1
730 K=K+S(X)+1
740 S(X)=S(X)+1
750 T(I)=T(I)+1
760 PRINT"O.K. IL PUNTEGGIO E' ";K
770 GOTO830
780 K=K-(10-S(X))
790 PRINT
800 PRINT"GIA' USATA ! PROVA ANCORA"
810 PRINT
820 GOTO620
830 NEXTZ
840 PRINT
850 PRINT
860 PRINT"E' TUTTO !"
870 PRINT
880 PRINT"IL TUO PUNTEGGI FINALE E' ";K
890 PRINT:PRINT:PRINT
900 INPUT"VUOI GIOCARE ANCORA (S/N)";Y$
910 IFY$="S"THEN20
920 END

READY.
```

LA DANZA DELLE PALLE

Questo programma grafico vi mostra un certo numero di palle che danzano sullo schermo. Esso vuole dimostrare quanto segue:

1) Animazione = le palle si muovono intorno allo schermo.

2) Programmabilità = potete programmare il movimento di ogni palla.

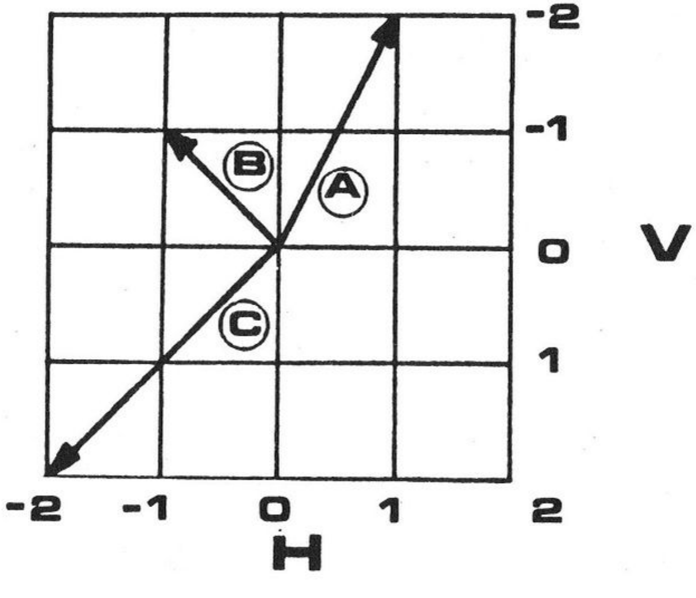
3) Multidimensionalità = possono essere presenti più palle, ognuna con il suo movimento. Per realizzare questo il programma usa una matrice a tre dimensioni.

Il programma chiede per prima cosa quante palle devono danzare, poi per ognuna chiede:

A) il numero dei movimenti che ogni palla deve fare;

B) gli spostamenti orizzontale e verticale per ogni movimento.

Il programma vi mostrerà poi la danza delle palle.



Il movimento (A) è 1,2

Il movimento (B) è −1,−1

Il movimento (C) è −2,2

Se desiderate che la palla stia ferma al suo posto date 0,0.

```
10 REM 18. LA DANZA DELLE PALLE
20 PRINT"♫"
30 LO=1024:HI=2023:WI=40:D=54272
40 L=HI-LO:TK=81
50 INPUT"QUANTE PALLE (FINO A 9)";N
60 DIMS(N,10,2),T(N),P(N),C(N)
70 FORI=1TON:P(I)=INT(RND(1)*L)+LO
80 PRINT
90 PRINT"QUANTI MOVIMENTI PER LA PALLA N.";I
100 INPUTT(I)
110 FORJ=1TOT(I)
120 PRINT
130 PRINT"COMUNICA IL MOVIMENTO";J;" PER LA PALLA";I
140 INPUTS(I,J,1),S(I,J,2)
150 NEXTJ:NEXTI
160 PRINT"♫"
170 FORI=1TON:C(I)=C(I)+1
180 IFC(I)>T(I)THENC(I)=1
190 NP=P(I)+S(I,C(I),1)+WI*S(I,C(I),2)
200 IFNP>HITHENNP=NP-L
210 IFNP<LOTHENNP=NP+L
220 POKENP,TK:POKEP(I),32:POKENP+D,6+I
230 P(I)=NP
240 NEXTI
250 GOTO170

READY.
```

SPACE SHUTTLE

State cercando di ritornare sulla terra a bordo di una navetta spaziale.

Sono dati: carburante, velocità e distanze. Il giocatore deve determinare l'entità della spinta necessaria per un atterraggio morbido.

Dopo ogni punto interrogativo dovete comunicare un numero fra 0 e 500.

```
10 REM 19. SPACE SHUTTLE
20 PRINT"⊐"
30 PRINT"****************************************"
40 PRINT
50 PRINT"HAI APPENA LASCIATO UN SATELLITE"
60 PRINT"E STAI SCENDENDO PER TORNARE SULLA TERRA"
70 PRINT"▒DOPO OGNI ? SCRIVI IL COMANDO DI SPINTA"
80 PRINT"DEL RAZZO (E' UN NUMERO DA 0 A 500)"
90 PRINT"*OGNI UNITA'=200 NEWTONS=20 KG DI SPINTA"
100 PRINT"--------BUONA FORTUNA--------"
110 PRINT:PRINT
120 PRINT"****************************************"
130 PRINT
150 PRINT"TEMPO  VELOC    ALT    CARB  SPINTA"
160 PRINT"SEC    M/SEC    KM     KG     N/100"
170 PRINT
180 H=110
190 H=INT((H-70)*RND(1)+70)
200 S=5000
210 E=10000
220 T=0
230 G=2.4
240 GOTO260
250 V=-100
260 V=INT(100*V+.5)/100:H=INT(100*H+.5)/100
263 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";
265 PRINTT;TAB(6);V;TAB(15);H;TAB(22);E;TAB(30);"F=";
270 INPUTF
280 IFABS(F-250)>251THEN310
290 IFF>ETHEN330
300 GOTO350
310 PRINT"SPINTA NON AMMESSA --RIPETI"
320 GOTO260
330 PRINT"HAI SOLO";E;"UNITA' DI CARBURANTE--RIPETI"
340 GOTO260
350 E=E-F
360 V1=V
370 V2=2000*F/(S+E)-10*I
380 V=V+V2
390 D=(V1+V2/2)*10
400 H=H+D/1000
410 IFH<.01THEN500
420 IFE<=0THEN450
430 T=T+10
440 GOTO260
450 PRINT"SEI RIMASTO SENZA CARBURANTE A";H:PRINT"CHILOMETRI DI ALTEZZA"
460 PRINT"TI FARA' PIACERE SAPERE CHE STAI"
470 PRINT:PRINT
480 GOTO600
490 F=0
500 K=0
510 K=K+.1
```

```
520 V1=1
530 V2=20*F/(S+E)-.1*I
540 V=V+V2
550 H=H-(V1+V2/2)/1000
560 PRINT"ATTERRAGGIO IN CIRCA";K;"SECONDI"
570 PRINT"ASPETTA":PRINT:PRINT
580 IFV>=-5THEN640
590 IFV>=-15THEN670
600 PRINT"PRECIPITANDO ALLA VELOCITA' DI=";V:"M/SEC"
610 PRINT"CIOE' A";V*3.2*3600/5280;"MPH--OH RAGAZZO !"
620 PRINT"RESTITUISCI LA TUA LICENZA DI VOLO"
630 END
640 PRINT"UN ATTERRAGGIO MERAVIGLIOSO A V=";V*3.2*3600/5280;"MPH"
650 PRINT"HAI ANCORA";E;"KG DI CARBURANTE"
660 END
670 PRINT"CLUNK--RATTLE--RATTLE--SQUINCH"
680 PRINT"V= ";V*3.2*3600/5280;"MPH : CIRCA, MA SEI SOPRAVVISSUTO"
690 END

READY.
```

INTRODUZIONE AI NUMERI BINARI

Per i capitoli successivi, dove avremo direttamente a che fare con i registri del computer, dobbiamo conoscere qualche cosa sui numeri binari ed esadecimali.

Come tutti ormai sanno, i computer capiscono soltanto i numeri binari, che possono avere solamente due posizioni: 0 oppure 1. Nel vostro COMMODORE 64, come nella maggior parte dei Personal Computer, otto di questi singoli numeri binari (chiamati bit) sono raggruppati insieme in una unità chiamata byte. Un byte può assumere ad esempio la seguente forma:

1010 1010

Un byte può avere fino a un totale di 256 diverse combinazioni di 1 e 0. La combinazione con il valore più basso è

0000 0000

La combinazione con il valore più alto è

1111 1111

Quello che abbiamo appena visto è la notazione binaria di un numero, ma poichè il linguaggio BASIC del vostro computer lavora solamente con numeri decimali, dobbiamo trasformare i numeri da base 2 a base 10. Questo si può fare come segue. Ogni posizione in un numero binario ha un moltiplicatore, che nell'ordine vale:

128 64 32 16 8 4 2 1

Questo significa che, per esempio, un 1 nella parte più a sinistra di un byte rappresenta un valore decimale di 128.

Per ottenere il valore decimale totale di un numero binario dobbiamo sommare tutti i moltiplicatori delle posizioni che contengono un 1. Per esempio:

```
 1    1    1    1     1    1    1    1
 :    :    :    :     :    :    :    :
128  +64  +32  +16   +8   +4   +2   +1   =   255
```

Oppure

```
1    0    1    0    1    0    1    0
:    :    :    :    :    :    :    :
128       +32       +8        +2        =   170
```

Poichè lavorare con così tante cifre è talvolta difficoltoso, c'è un altro modo di rappresentare i numeri binari: la notazione esadecimale. Mediante questa notazione quattro cifre binarie sono rappresentate da un numero esadecimale.

Quattro numeri binari possono avere 16 diverse combinazioni: queste combinazioni, i loro equivalenti esadecimali e decimali sono elencati qui di seguito:

BINARIO	ESADECIMALE	DECIMALE
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

Per i numeri esadecimali con più di una cifra dobbiamo conoscere i moltiplicatori di ogni singola posizione. Per un numero di quattro cifre essi sono i seguenti:

4096   256   16   0

Abitualmente i numeri esadecimali vengono scritti facendoli precedere dalla lettera $ (dollaro), come per esempio:

$D400

Se desideriamo convertire questo numero in un numero decimale dobbiamo moltiplicare ogni posizione per il rispettivo moltiplicatore e fare il totale dei prodotti così ottenuti

```
$ D        4        0        0
  :        :        :        :
13*4096  +4*256                  = 54272
```

Ecco il listato di un programma che converte i numeri binari nel loro equivalente esadecimale.

```
100 REM 20. CONVERSIONE DI NUMERI
110 PRINT"□"
120 PRINT"NUMERO BINARIO        : ";
130 FORX=1TO8
140 GETA$(X):IFA$(X)=""THEN140
150 IFA$(X)="1"ORA$(X)="0"THEN170
160 GOTO140
170 PRINTA$(X);
180 NEXTX
190 PRINT:PRINT
200 T=0
210 FORX=0TO7
220 IFVAL(A$(X+1))=1THENT=T+2↑(7-X)
230 NEXTX
240 PRINT"EQUIVALENTE DECIMALE:";T
250 PRINT:PRINT
260 GOTO120

READY.
```

A volte possiamo desiderare che un bit in un registro diventi 1 o 0, mentre vogliamo che gli altri bit restino come sono (non conosciamo il loro valore in quel momento). Ci sono due comandi che possiamo usare in questo caso: il comando AND e il comando OR. Se desideriamo inserire uno 0 usiamo AND con un numero binario che ha sette cifre 1 e una cifra 0, ponendo lo 0 nella posizione che desideriamo esso abbia nel risultato. Se desideriamo inserire un 1 usiamo OR con un numero binario che ha sette cifre 0 e una cifra 1, ponendo l'1 nella posizione che desideriamo esso abbia nel risultato.

Esempio:

```
POKE 53272, PEEK(53272)AND(255—128)
```

Questa istruzione farà assumere il valore 0 al bit più alto dell'indirizzo 53272, lasciando inalterati tutti gli altri. Il numero decimale 255 in notazione binaria è 1111 1111 e se noi sottraiamo 128 dal tal numero esso diventa 01111111, poichè il valore del bit più alto è 128. Infine l'istruzione

```
POKE 53272,PEEK(53272)OR1
```

metterà un 1 nel bit più basso dell'indirizzo 53272 lasciando immutati tutti gli altri.

CADUTA LIBERA E LANCIO VERSO L'ALTO

Molti giochi contengono movimenti di lancio o sparo di oggetti. Le traiettorie di un oggetto possono essere calcolate con equazioni matematiche e queste configurazioni di vari movimenti possono essere raggruppate in cadute libere o lanci secondo curve.

In questo capitolo simuliamo la caduta libera ed il movimento secondo un certo angolo usando il comando POKE per rappresentare un carattere grafico. Noi possiamo pensare alla memoria dello schermo come ad un sistema di coordinate. Poiché la memoria dello schermo ha il suo indirizzo più basso nell'angolo in alto a sinistra e un normale sistema di coordinate ha la sua origine nell'angolo in basso a destra, dobbiamo convertire i valori della coordinata Y togliendo il valore calcolato dall'indirizzo alla sommità del video. Le equazioni di un oggetto in caduta libera in un sistema di coordinate sono

V=G*T+VO
Y=G/2*T*T+VO*T+YO

dove G è la accelerazione di gravità, T è il tempo in secondi e VO è la velocità iniziale al tempo T=O. Y è l'altezza e YO è l'altezza iniziale al tempo T=O.

Nel primo esempio simuliamo una caduta libera da un'altezza di 2000 metri: lasciamo cadere un solido con una velocità iniziale VO=O.

```
100 REM 21. CADUTA LIBERA
110 Y0=2000:V0=0
120 G=-9.81:T=0
130 H=Y0
140 S=175/Y0
150 PRINT"♥"
160 Y=G/2*T*T+V0*T+Y0
170 LOC=1040+40*INT((24-(Y*S/8))):COL=55312+40*INT((24-(Y*S/8)))
180 POKE LOC,81:POKE COL,4
190 IF Y<0 THEN STOP
200 T=T+1
210 GOTO 160

READY.
```

I valori iniziali sono V0=0 e Y0=2000.

Dopo ogni calcolo della nuova altezza il tempo è incrementato di una unità. Per cambiare la velocità del movimento si può cambiare l'incremento

del tempo. Ora cambiamo il programma in modo tale che il movimento dell'oggetto sia verso l'alto. Alla velocità iniziale potrete assegnare un valore compreso tra 100 e 200 senza che l'oggetto lasci lo schermo.

```
100 REM 22. LANCIO VERSO L'ALTO
110 PRINT"□"
120 INPUT"VELOCITA' INIZIALE V0 (100-200)";V0
130 G=-9.81:T=0
140 Y0=0:H=Y0
150 S=175/2000
160 Y=G/2*T*T+V0*T+Y0
170 COL=55312+40*INT((24-(H*S/8)))
180 POKE COL,6
190 LOC=1040+40*INT((24-(Y*S/8))):COL=55312+40*INT((24-(Y*S/8)))
200 POKE LOC,81:POKE COL,4
210 H=Y
220 IF Y<0 THEN STOP
230 T=T+1
240 GOTO 160

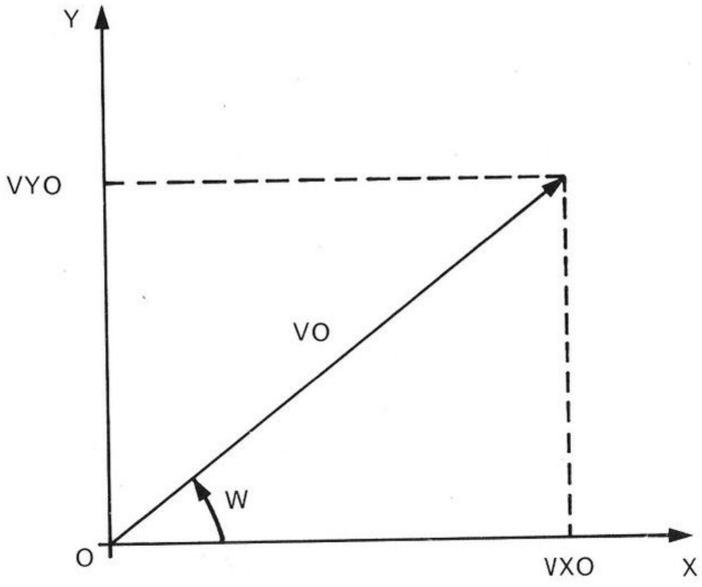
READY.
```

IL MOVIMENTO IN DIREZIONE OBLIQUA

Al precedente movimento sovrapponiamo ora un movimento nella direzione X con velocità costante pari a VX=C. La posizione nella direzione X è costante dalla equazione X=C*T+V0. Abbiamo le seguenti quattro equazioni:

VY=—G*T+VY0
VX=VX0
Y=—G/2*y=—G/2*T*T+VY0*T+Y0
X=VX0X=VX0*T+X0

Tutte le parabole dovrebbero partire dall'angolo in basso a sinistra alle coordinate X=0 e Y=0, quindi i valori iniziali di X0 e Y0 sono nulli. Le condizioni di partenza sono: l'angolo e la velocità iniziale. La figura sottostante illustra come la velocità V0 si scompone nella velocità iniziale VX0 in direzione X e VYO in direzione Y a seconda dell'angolo iniziale di tiro W.



Abbiamo quindi VXO=VO*cosW e VY0=V0*sinW.

Il programma listato sotto disegna il movimento di un oggetto lanciato in direzione obliqua. Per illustrare i vari gradi della potenza del lancio potrete introdurre numeri da 100 200.

```
100 REM 23. PARABOLA
110 PRINT"J"
120 X0=0:Y0=0
130 PRINT"J":INPUT"ANGOLO      ";W
140 INPUT"FORZA       ";V0
150 W=W*2*3.1416/360
160 VX=V0*COS(W)
170 VY=V0*SIN(W)
180 G=9.81
190 T=0:H=0:L=0
200 Y=-G/2*T*T+VY*T
210 IF Y<0 THEN FOR D=1 TO 500:NEXT D:GOTO 130
220 X=VX*T
230 LOC=1024+40*INT(24-Y/50)+INT(X/50)
240 COL=55296+40*INT(24-Y/50)+INT(X/50)
250 IF LOC<1024 THEN T=T+1:GOTO 200
260 POKE LOC,81:POKE COL,4
270 L=X:H=Y
280 T=T+1
290 GOTO 200

READY.
```

LA GRAFICA CON IL COMMODORE 64

Normalmente la memoria dello schermo del COMMODORE 64 parte dalla locazione 1024 ($0400) e va fino alla locazione 2023 ($07E7), per un totale di 1000 differenti posizioni (40 caratteri per 25 righe).

Ogni locazione può avere un valore da 0 a 255 per rappresentare i diversi caratteri presi dalla tabella dei caratteri.

In corrispondenza con la memoria dello schermo ci sono 1000 locazioni di memoria che vanno dall'indirizzo 55296 fino al 56295 (da $D800 a $DBE7) e che definiscono il colore del carattere nel corrispondente indirizzo della memoria dello schermo. L'insieme dei caratteri usati può essere sia quello già assemblato nella macchina, memorizzato nella ROM del vostro COMMODORE 64, sia uno qualunque creato da voi e che memorizzerete nella RAM. Per comunicare al computer quale insieme di caratteri deve utilizzare, occorre posizionare certi bit in un registro di controllo localizzato all'indirizzo 53272. Questi sono i bit 1,2 e 3, mentre il bit 0 è ignorato in questo registro di controllo. Se i quattro bit più bassi rappresentano un valore di 0 (o 1), come set di caratteri vengono utilizzati i 2K di memoria più bassi ($0000-$07FF), mentre un valore di 2 (o 3) seleziona i successivi 2K ($0800-$0FFF).

Gli altri numeri possibili sono:

```
4 (or 5) -> $1000-$17FF (normale)
6 (or 7) -> $1800-$1FFF (lettere minuscole)
8 (or 9) -> $2000-27FF
10 (or 11) -> $2800-$2FFF
12 (or 13) -> $3000-$37FF
14 (or 15) -> $3800-$3FFF
```

Ogni carattere occupa sempre 64 bit pari a 8 byte. Per illustrare l'uso del set di caratteri provate il programma listato più avanti. Supponiamo di volere un set di caratteri simile a quello regolare, ma leggermente diverso, per esempio desideriamo che ogni lettera sia sottolineata. Questo può essere fatto nel seguente modo.

COLOR-RAM

												+ 10										+ 20										+ 30									
Adress		0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
55296																																									
55336																																									
55376																																									
55416																																									
55456																																									
55496																																									
55536																																									
55576																																									
55616																																									
55656																																									
55696																																									
55736																																									
55776																																									
55816																																									
55856																																									
55896																																									
55936																																									
55976																																									
56016																																									
56056																																									
56096																																									
56136																																									
56176																																									
56216																																									
56256																																									

SCREEN-RAM

												+10										+20										+30									
Adress		0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
1024																																									
1064																																									
1104																																									
1144																																									
1184																																									
1224																																									
1264																																									
1304																																									
1344																																									
1384																																									
1424																																									
1446																																									
1504																																									
1544																																									
1584																																									
1624																																									
1664																																									
1704																																									
1744																																									
1784																																									
1824																																									
1864																																									
1904																																									
1944																																									
1984																																									

Trasferiamo nella RAM del computer il set di caratteri che normalmente è memorizzato nella ROM, eseguiamo le nostre variazioni e poi faremo sempre riferimento a questo gruppo di caratteri. Il programma esegue il trasferimento di un gruppo di 32 caratteri. Per fare questo il programma deve disattivare un interrupt, il che avviene alla linea 110, dopo il trasferimento l'interrupt è attivato di nuovo (linea 170).

La linea 120 scollega l'I/0 (INPUT/OUTPUT) e connette la ROM dei caratteri, la linea 160 ritorna al modo normale. La linea 180 posiziona il computer dal normale set di caratteri contenuti nella ROM a quello appena creato nella RAM.

Non appena avrete fatto partire il programma e il computer avrà eseguito la linea 180, noterete che tutte le lettere sono rappresentate normalmente, mentre al posto dei numeri appariranno dei simboli confusi. Questo accade perché noi abbiamo copiato soltanto i primi 32 caratteri del set e quando il computer cerca di costruire l'immagine di un numero trova nella RAM dei numeri casuali. Per trasferire più caratteri dovete semplicemente cambiare il numero 255 in 511 nella linea 130 e dare un nuovo RUN al programma.

```
READY.

100 REM 24. TRANSFERIMENTO DEL SET DEI CARATTERI
110 POKE56334,PEEK(56334)AND254
120 POKE1,PEEK(1)AND251
130 FORI=0TO255
140 POKE12288+I,PEEK(53248+I)
150 NEXTI
160 POKE1,PEEK(1)OR4
170 POKE56334,PEEK(56334)OR1
180 POKE53272,(PEEK(53272)AND240)+12
190 END

READY.
```

Per ottenere i caratteri sottolineati usiamo il programma elencato sotto, che semplicemente scrive un 255 (che rappresenta un trattino continuo) nell'ultimo byte, l'ottavo, di ogni carattere.

```
READY.

199 REM 25. CARATTERI SOTTOLINEATI
200 FORX=0TO64
210 POKE12288+7+X*8,255
220 NEXTX

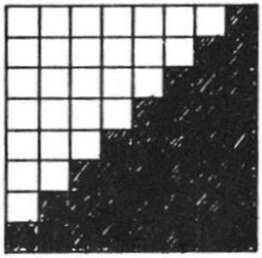
READY.
```

Ora noi desideriamo trasferire un paesaggio di nostra creazione sullo schermo usando un set di caratteri espressamente progettato. Pensiamo di

disegnare una montagna: ci servono tre diversi caratteri grafici, uno per rappresentare la salita uno per la discesa e uno continuo per riempire il disegno.

Assegneremo il carattere raffigurante la salita al tasto A, quello di riempimento al tasto B e quello per la discesa al tasto C. I caratteri usati sono creati facilmente.

A



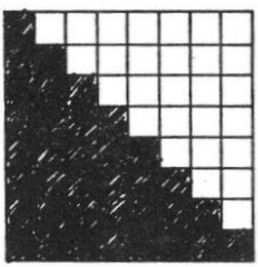
1
3
7
15
31
63
127
255

B



255
255
255
255
255
255
255
255

C



128
192
224
240
248
252
254
255

Il programma listato crea un nuovo set di caratteri partendo dalla locazione 12288. Solamente i tasti 'A', 'B' e 'C' hanno un significato in questo nuovo gruppo di caratteri e poiché i caratteri usati sono il secondo, il terzo e

il quarto del set medesimo, dobbiamo identificare un carattere (il primo carattere di ogni gruppo è assegnato al tasto @).

Questo è il motivo per cui nella linea 130 un 8 è aggiunto all'indirizzo. La linea 150 innesca il nuovo set di caratteri, ma non appena questa linea è eseguita voi dovreste vedere un po' di confusione sullo schermo. Ogni volta che voi premete 'A', 'B' o 'C' i simboli desiderati dovrebbero apparire. Voi potreste infatti andare avanti e disegnare ora una montagna usando i tasti di controllo del cursore e i tasti 'A', 'B', 'C'. Premere RETURN causerebbe un SINTAX ERROR:
per ritornare al modo normale di rappresentazione dovete premere contemporaneamente i tasti RUN/STOP e RESTORE.

```
READY.

100 REM 26. NUOVI SIMBOLI PER A B C
110 FORX=0TO23
120 READA
130 POKE12288+8+X,A
140 NEXTX
150 POKE53272,(PEEK(53272)AND240)+12
200 DATA1,3,7,15,31,63,127,255
210 DATA255,255,255,255,255,255,255,255
220 DATA128,192,224,240,248,252,254,255

READY.
```

A questo punto non dovreste avere problemi a mettere insieme i vostri simboli per rappresentare una montagna. Il programma listato sotto fa proprio questo e in più pittura la montagna in tre diversi colori scrivendo dei numeri negli indirizzi corrispondenti della RAM dei colori (linee da 400 a 480).

```
100 REM 27. MONTAGNE
110 FORX=0TO23
120 READA
130 POKE12288+8+X,A
140 NEXTX
150 POKE53272,(PEEK(53272)AND240)+12
200 DATA1,3,7,15,31,63,127,255
210 DATA255,255,255,255,255,255,255,255
220 DATA128,192,224,240,248,252,254,255
230 PRINT"♥"
300 PRINT"[illegible]";
310 PRINT"     AC"
320 PRINT"    ABBC"
330 PRINT"   ABBBBC"
340 PRINT"  ABBBBBBC"
350 PRINT" ABBBBBBBBC"
360 PRINT"ABBBBBBBBBBC"
400 FORX=55776TO55776+39
410 POKEX,1:REM WHITE
420 NEXTX
430 FORX=55816TO55816+79
440 POKEX,12:REM MED GREY
450 NEXTX
460 FORX=55896TO55896+119
470 POKEX,5:REM GREEN
480 NEXTX
490 GOTO490

READY.
```

Il passo successivo è quello di creare simboli diversi e di costuire un «paesaggio». Il programma successivo vi mostra qualche cosa a questo proposito, ma ora potrete sbrigliare la vostra fantasia e creare varie immagini di sfondo.

```
100 REM 28. PAESAGGIO
110 FORX=0TO47
120 READA
130 POKE12288+8+X,A
140 NEXTX
150 POKE53272,(PEEK(53272)AND240)+12
160 FORX=0TO7
170 POKE12288+32*8+X,0:REM CANCELLA
180 NEXTX
200 DATA1,3,7,15,31,63,127,255:REM 'A'
210 DATA255,255,255,255,255,255,255,255:REM 'B'
220 DATA128,192,224,240,248,252,254,255:REM 'C'
230 DATA1,1,15,15,1,1,1,1:REM 'D'
240 DATA128,128,240,240,128,128,128,128:REM 'E'
250 DATA255,255,195,195,195,195,255,255:REM 'F'
290 PRINT"{CLR}"
300 PRINT"{HOME}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}";
310 PRINT"                               AC"
320 PRINT"     DE                       ABBC"
330 PRINT"     AC          AC          ABBBBC"
340 PRINT"    ABBC        ABBC        ABBBBBBC"
350 PRINT"   ABBBBC      ABBBBC      ABBBBBBBBC"
360 PRINT"  ABBBBBBC    ABBBBBBC    ABBBBBBBBBBC"
370 PRINT" ABBBBBBBBC  ABBBBBBBBC  ABBBBBBBBBBBBC"
380 PRINT"ABBBBBBBBBBCABBBBBBBBBBCABBBBBBBBBBBBBBC"
400 PRINT"{HOME}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}";
410 PRINT"     ABBBBBBBC"
420 PRINT"      FFFFFFF            ABBBBBBC"
430 PRINT"      FFFFFFF             FFFFFF"
440 PRINT"      FFFFFFF             FFFFFF"
450 PRINT"      FFFFFFF             FFFFFF"
460 PRINT"      FFFFFFF             FFFFFF"
470 PRINT"      FFFFFFF             FFFFFF"
480 PRINT:PRINT"BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB";
490 PRINT"BBBBBBBBBBBBBBBBBBBB"
500 FORX=0TO119
510 POKEX+55376,1:REM NEVE
520 NEXTX
530 FORX=0TO119
540 POKEX+55496,12:REM ROCCE
550 NEXTX
560 FORX=0TO79
570 POKEX+55616,5:REM ERBA
580 NEXTX
590 FORX=0TO39
600 POKEX+55816,2:REM TETTO
610 NEXTX
620 FORX=0TO239
630 IFX>20ANDX<40THENPOKE55856+X,2:NEXTX
640 POKEX+55856,7:REM CASA
650 NEXTX
660 FORX=0TO79
670 POKEX+56136,15:REM STRADA
680 NEXTX
700 POKE 55416+5,9:POKE55416+6,9:REM INCROCIO
710 GOTO710

READY.
```

GRAFICA IN ALTA RISOLUZIONE

Con il COMMODORE 64 avrete la possibilità di disegnare con la grafica ad alta risoluzione, in cui ogni punto in una matrice di 320*200 può essere controllato separatamente.

In grafica ad alta risoluzione 8 bit formano un byte, 8 byte definiscono uno spazio (l'area che nel modo "carattere" è occupata per l'appunto da un carattere), 40 spazi definiscono una linea e 25 linee definiscono l'intero schermo. Quindi abbiamo un totale di 8 *40*25=8000 byte o 8*8000=64000 punti sullo schermo. Il byte con l'indirizzo più basso (byte 0) è quello che definisce la prima linea nello spazio dell'angolo superiore sinistro. Il byte 1 definisce la seconda linea di questo spazio, il byte 7 definisce la linea di fondo del primo spazio. Il byte 8 definisce la prima linea del secondo spazio e così via. Il byte 319 definisce l'ultima linea dello spazio 40. L'ultimo byte sullo schermo è quello con il numero 7999, situato nell'ultima linea dello spazio nell'angolo più basso a destra del video.

L'indirizzo di base per la grafica ad alta risoluzione può essere posto nella locazione 53272. Il seguente comando posiziona l'indirizzo di base a 8192 ($2000), per esempio

```
POKE53272, PEEK (53272) OR8
```

Questo assegnerà tutti gli indirizzi da 8192 a 16383 alla grafica ad alta risoluzione. Il modo grafico ad alta risoluzione è attivato posizionando il bit 5 nel registro di indirizzo 53265 a 1 con il seguente comando:

```
POKE53265,PEEK(53265)OR32
```

Dopo l'esecuzione di questi due comandi lo schermo sarà riempito con simboli grafici casuali. Questo avviene perché la partizione assegnata alla grafica ad alta risoluzione è riempita con dati casuali, ora interpretati come segni grafici.

Per cancellare questa parte di memoria dovrete digitare:

```
FORI=8192TO16383:POKEI,O:NEXTI
```

Il colore dello sfondo e i punti con alta risoluzione grafica sono definiti dal contenuti di quella che normalmente è la memoria dello schermo (indirizzi da 1024 a 2023). I quattro bit superiori di questi indirizzi ora definiscono il colore di un punto nel corrispondente spazio sullo schermo; i quattro bit inferiori definiscono il colore dello sfondo. Se avete deciso il medesimo colore su tutto lo schermo, dovete digitare il seguente comando:

```
FORI=1024TO2023:POKEI,16*DC+BC:NEXTI
```

dove DC è il colore dei punti e BC è il colore dello sfondo (entrambi i valori possono variare da 0 a 15).

Per indirizzare un singolo punto in alta risoluzione dobbiamo dare le coordinate X e Y e convertirle nell'indirizzo di un byte. Il valore del byte definisce quale bit a quell'indirizzo deve essere attivato. Le formule per questi calcoli sono:

numero della fila=coordinata Y/8
carattere=coordinata X/8
numero della linea=coordinata Y AND 7
bit=7 − (X AND 7)
indirizzo del byte=indirizzo della base+320*numero della fila+8*carattere+numero della linea

All'interno dell'indirizzo del byte dobbiamo assegnare un valore con l'istruzione POKE, riferendoci alla posizione del bit desiderato. In quel byte e allo scopo di lasciare inalterati gli altri bit ad esso appartenenti e già posizionati, usiamo il seguente comando:

```
POKE indirizzo del byte, PEEK (indirizzo del byte) OR2 bit
```

Il programma listato a fianco esegue tutti questi calcoli. Voi dovete inserire le coordinate di un punto e il programma lo visualizzerà sullo schermo alla posizione corrispondente.

Potrete poi estendere il programma per disegnare linee, ecc.

Se scriviamo una subroutine al di fuori di questo programma, possiamo calcolare le coordinate (ad esempio di una curva) e poi saltare a questa subroutine per disegnare i punti calcolati.

```
100 REM 29. GRAFICA AD ALTA RISOLUZIONE
110 POKE53272,21OR8:POKE53265,27OR32
120 PRINT"⊐";:BA=8192
130 FORC=BATOBA+7999
140 POKEC,0
150 NEXTC
160 POKE53265,27:POKE53272,21
170 INPUT"COORDINATA X  (0-319)";X
180 INPUT"COORDINATA Y  (0-199)";Y
190 POKE53272,21OR8
200 POKE53265,27OR32
210 RO=INT(Y/8)
220 CH=INT(X/8)
230 LI=YAND7
240 BI=7-(XAND7)
250 BY=BA+RO*320+CH*8+LI
260 POKEBY,PEEK(BY)OR2↑BI
270 FORD=1TO2000:NEXTD
280 POKE53265,27:POKE53272,21
290 PRINT"▨";:GOTO 170
READY.
```

Il programma successivo vi permette di disegnare una retta uscente dal punto di coordinate X1,Y1 fino al punto di coordinate X2,Y2: questo programma quindi sostituisce il comando DRAW TO (disegna fino a) che manca nel COMMODORE 64.

```
5 REM 30. SIMULAZIONE DELL'ISTRUZIONE DRAW TO (DISEGNA FINO A)
10 INPUT"ASCISSA  INIZIALE X1";X1
20 INPUT"ORDINATA INIZIALE Y1";Y1
30 INPUT"ASCISSA  FINALE   X2";X2
40 INPUT"ORDINATA FINALE   Y2";Y2
42 POKE 53265,59
44 POKE 53272,25
45 FOR V=1024 TO 2023
46 POKE V,16
47 NEXT V
48 FOR T=8192 TO 16383
49 POKE T,0
50 NEXT T
55 DX=X2-X1:DY=Y2-Y1
57 IF ABS(DY)<ABS(DX) THEN 64
58 FOR YL=Y1 TO Y2 STEP SGN(DY)
59 X=DX/DY*YL+X1
60 Y=YL
61 GOSUB 200
62 NEXT YL
63 GOTO 300
64 FOR XL=X1 TO X2 STEP SGN(DX)
65 X=XL
66 Y=DY/DX*XL+Y1
67 GOSUB 200
68 NEXT XL
69 GOTO 300
200 YP=INT(Y/8)
210 XP=INT(X/8)
220 A1=(YP*40+XP)*8
230 AY=Y-8*YP+A1
240 AA=8192
250 R=X-8*XP
260 M=2↑(7-R)
270 I=PEEK(AY+AA)
280 POKE AY+AA,I OR M
290 RETURN
300 FOR T=1 TO 5000:NEXT T
310 POKE 53265,27:POKE 53272,21
320 END
READY.
```

Ecco la subroutine per disegnare un singolo punto in alta risoluzione: dovete assegnare un valore da 0 a 319 alla variabile X e un valore da 0 a 199 alla variabile Y e saltare alla subroutine con il comando GOSUB

```
FILA=INT (Y/8)
CARATTERE=INT (X/8)
LINEA=Y AND 7
BIT=7-(X AND 7)
BYTE=BASE+FILA*320+CARATTERE*8+LINEA
POKE BYTE,PEEK (BYTE) OR 2 ^BIT
RETURN
```

Segue ora un altro programma che utilizza questa subroutine: si tratta della versione in alta risoluzione del programma già listato precedentemente in questo libro per simulare il movimento in direzione obliqua di un oggetto

```
100 REM 31. MOVIMENTO IN DIREZIONE OBLIQUA (IN ALTA RISOLUZIONE)
110 PRINT"♥"
120 INPUT"INSERISCI L'ANGOLO  (0-90)";W
130 INPUT"INSERISCI LA FORZA  (0-80)";V0
140 W=W*2*3.1416/360
145 PRINT:PRINT"PULISCO L'AREA DELL'ALTA RISOLUZIONE..."
150 FORI=0TO7999
160 POKE8192+I,0
170 NEXTI
175 PRINT"PULISCO L'AREA DEL COLORE..."
180 FORI=0TO999
190 POKE1024+I,16
200 NEXTI
210 BA=8192:POKE53272,21OR8:POKE53265,27OR32
220 VX=V0*COS(W)
230 VY=V0*SIN(W)
240 G=9.81
250 T=0:H=0:L=0
260 Y=-G/2*T*T+VY*T:Y=200-Y
270 X=VX*T
280 IFY>200THEN280
290 IFY<0THEN310
300 GOSUB1000
310 L=X:H=Y
320 T=T+.1
330 GOTO260
1000 REM SUBROUTINE DI TRACCIAMENTO DEL GRAFICO
1010 RO=INT(Y/8)
1020 CH=INT(X/8)
1030 LI=YAND7
1040 BI=7-(XAND7)
1050 BY=BA+RO*320+CH*8+LI
1060 POKEBY,PEEK(BY)OR2↑BI
1070 RETURN

READY.
```

Il programma successivo mostra come è possibile rappresentare una funzione matematica utilizzando la grafica ad alta risoluzione.

```
100 REM 32. GRAFICO DI UNA FUNZIONE MATEMATICA
110 FORI=0TO7999
120 POKE8192+I,0
130 NEXTI
140 FORI=0TO999
150 POKE1024+I,16
160 NEXTI
170 BA=8192:POKE53272,21OR8:POKE53265,27OR32
180 FORI=0TO360STEP3
190 D=I*π/180
200 Y=30*SIN(D):Y=-Y+100
210 X=I/1.5
220 GOSUB1000
230 NEXTI
240 GOTO240
1000 REM SUBROUTINE DI TRACCIAMENTO DEL GRAFICO
1010 RO=INT(Y/8)
1020 CH=INT(X/8)
1030 LI=YAND7
1040 BI=7-(XAND7)
1050 BY=BA+RO*320+CH*8+LI
1060 POKEBY,PEEK(BY)OR2↑BI
1070 RETURN

READY.
```

LA FUNZIONE MULTICOLORE

Nella grafica ad alta risoluzione potete controllare ogni punto sullo schermo. Ogni bit sullo schermo può valere uno, e quindi rappresentare un punto nel colore definito nel registro dei colori corrispondente a quello spazio, o 0 e rappresentare uno spazio bianco mostrato nel colore dello sfondo.

Se operate nel modo multicolore potete avere quattro diversi colori per ogni punto. Poiché sono necessari due bit per definire un'uscita di quattro possibili combinazioni, due bit nella direzione orizzontale sono visti come uniti insieme.

Questo naturalmente dà origine ad una risoluzione in senso orizzontale, che è solamente la metà di quella normale, poiché un punto in questo modo di funzionare del computer è due volte più largo di quello rappresentato nella visione normale. Per attivare il modo multicolore il bit 4 nell'indirizzo 53270 ($DO16) deve essere posto a 1 impiegando il seguente comando

```
POKE53270,PEEK(53270)OR16
```

Dopo che avete attivato il modo multicolore con questo comando dovete ancora scegliere, per ogni spazio, tra modo di impiego multicolore e modo in grafica ad alta risoluzione e questo dipende dal numero che voi scrivete nella corrispondente locazione della RAM dei colori.

Se inserite, con il comando POKE, un numero da 0 a 7 nella memoria di colore, lo spazio corrispondente sullo schermo sarà visualizzato nella grafica ad alta risoluzione standard.

Ogni bit posizionato sarà mostrato nel colore definito nella particolare locazione della RAM di colore. Se voi inserite, con la POKE, un numero da 8 a 15 nella memoria di colore, allora lo spazio corrispondente sarà visualizzato in modo multicolore.

In dettaglio: se una coppia di bit è 00, allora nella corrispondente locazione è mostrato il colore dello sfondo numero zero, il normale colore dello schermo. La coppia di bit 01 è per il colore di sfondo numero uno, la coppia 10 è per il numero due e infine la coppia di bit 11 produce la visualizzazione

del colore specificato nella RAM di colore (valori da 0 a 7). Le locazioni per questi colori sono:

sfondo N. 0: 55281 ($D021)
sfondo N. 1: 55282 ($D022)
sfondo N. 2: 55283 ($D023)

SCROLLING REGOLARE

Il COMMODORE 64 vi permette di effettuare una traslazione regolare (scrolling) dell'intero schermo in direzione orizzontale oppure verticale, un pixel (unità grafica) per volta, fino a 7 pixel. Se desiderate avere un movimento più esteso dovrete procedere come segue.

Dopo uno scrolling di 7 pixel, chiamate un programma in linguaggio macchina (è scritto in linguaggio macchina per motivi di rapidità di esecuzione), che prima pone lo chermo nella sua posizione originale, poi muove l'intera memoria del video di un carattere o di una linea per volta a seconda se lo scrolling è orizzontale o verticale. Questo movimento è chiamato scrolling irregolare. Quando traslate orrizzontalmente dovete restringere lo schermo a 38 colonne con il seguente comando

POKE53265,PEEK(53265)AND247

Quando traslate verticalmente dovete restringere lo schermo a 24 righe con il seguente comando

POKE53270,PEEK(53270)AND 247

Questa procedura vi lascia spazio per partire con lo scrolling di nuovi dati. Dai tre bit inferiori di queste locazioni dipende la bontà dello scrolling.

I seguenti comandi eseguono un buon scrolling

POKE53270,(PEEK(53270)AND248) +X

per movimenti orizzontali, e

POKE53265,(PEEK(53265)AND248) +Y

per movimenti verticali. X e Y possono assumere valori da 0 a 7. Digitate e lanciate il seguente programma:

```
9 REM 33. SCROLLING (SPOSTAMENTO) DELLO SCHERMO
10 POKE53270,PEEK(53270)AND247
20 FORX=7TO0STEP-1
30 POKE53270,(PEEK(53270)AND248)+X
40 FORD=1TO100:NEXTD
50 NEXTX
60 GOTO20
READY.
```

Come vedrete, i caratteri mostrati sullo schermo trasleranno di 7 pixel verso sinistra poi il movimento ripartirà di nuovo.

Il seguente programma in linguaggio macchina riposiziona lo scrolling nella posizione originale, poi muove l'intera memoria di schermo di un carattere verso sinistra. La miglior locazione per il programma in linguaggio macchina è all'indirizzo 49152 ($C000), poiché questa area non è occupata da programmi in BASIC. Ecco il programma

```
$C000  A9 07          LDA #7
$C002  0D 16 D0       ORA $D016
$C005  8D 16 D0       STA $D016
$C008  A2 00          LDX #0
$C00A  BD 01 04    M1 LDA $0401,X ;1st line
$C00D  9D 00 04       STA $0400,X
$C010  BD 29 04       LDA $0429,X ;2nd line
$C013  9D 28 04       STA $0428,X
          .               .
          .               .
       BD C1 07       LDA $07C1,X ;25th line
       9D C0 07       STA $07C0,X
       E8             INX
       E0 28          CPX #40
       F0 03          BEQ RET
       4C 0A C0       JMP M1
       60         RET RTS
```

Il programma in BASIC listato più sotto inserisce questo programma in linguaggio macchina con le istruzioni POKE, esegue lo scrolling regolare, chiama la routine in linguaggio macchina con il comando SYS(49152) per lo scrolling irregolare e infine riparte di nuovo.

Con la grafica ad alta risoluzione il programma in linguaggio macchina per muovere l'intero schermo è diverso perché in quel modo di impiego devono essere spostati 8000 byte invece di 1000.

Nota: se state usando colori diversi sullo schermo dovete anche spostare i contenuti della RAM di colore nella stessa direzione della RAM di schermo. Un programma in linguaggio macchina è sempre necessario.

```
100 REM 34. SCROLLING (SPOSTAMENTO) DELLO SCHERMO VERSO SINISTRA
110 FORX=0TO168
120 READ Z
130 POKE49152+X,Z
140 NEXTX
150 POKE53270,PEEK(53270)AND247
160 POKE53270,(PEEK(53270)AND248)+7
170 FORP=6TO0STEP-1
180 POKE53270,(PEEK(53270)AND248)+P
200 NEXTP
210 SYS(49152)
220 GOTO170
240 DATA169,7,13,22,208,141,22,208
250 DATA162,0,189,1,4,157,0,4
260 DATA189,41,4,157,40,4,189,81,4,157,80,4
270 DATA189,121,4,157,120,4,189,161,4,157,160,4
280 DATA189,201,4,157,200,4,189,241,4,157,240,4
290 DATA189,25,5,157,24,5,189,65,5,157,64,5
300 DATA189,105,5,157,104,5,189,145,5,157,144,5
310 DATA189,185,5,157,184,5,189,225,5,157,224,5
320 DATA189,9,6,157,8,6,189,49,6,157,48,6
330 DATA189,89,6,157,88,6,189,129,6,157,128,6
340 DATA189,169,6,157,168,6,189,209,6,157,208,6
350 DATA189,249,6,157,248,6,189,33,7,157,32,7
360 DATA189,73,7,157,72,7,189,113,7,157,112,7
370 DATA189,153,7,157,152,7,189,193,7,157,192,7
380 DATA232,224,40,240,3,76,10,192,96

READY.
```

SPRITE

In questo capitolo parleremo di una caratteristica veramente speciale del COMMODORE 64: gli sprite.

Uno sprite è un oggetto grafico la cui forma e colore sono definibili dall'utente e che può essere facilmente mosso sullo schermo semplicemente comunicando al computer le coordinate del video, dove desiderate che esso sia visualizzato. Questo è il vantaggio della grafica con gli sprite sulla grafica convenzionale, dove voi dovete muovere i vostri oggetti grafici nella memoria, se volete cambiare ad essi di posizione sullo schermo. Quindi gli sprite grafici sono perfettamente adatti per programmare giochi con oggetti in rapido movimento.

Con il COMMODORE 64 voi potete rappresentare e muovere indipendentemente fino a otto sprite (dallo sprite 0 allo sprite 7). I registri che vi permettono di controllare questi sprite sono nelle locazioni di memoria che partono da $D000=53248. Il segno dollaro ($) che precede un numero significa che questo numero è in base esadecimale. Un numero senza nulla che lo precede è un normale numero in base dieci. D'ora in avanti assegneremo questo indirizzo alla variabile V. I registri hanno quindi indirizzi come V+6,V+23, ecc.

Prima che uno sprite appaia sullo schermo sono necessari molti passaggi. Dobbiamo introdurre le seguenti definizioni:

1) Gli sprite devono essere attivati. L'indirizzo per farlo è il V+21. Se desideriamo attivare lo sprite 0 dobbiamo memorizzare un 1 (con la POKE) in questo indirizzo, per lo sprite 7 è richiesto un 128. In generale il numero da memorizzare è pari a 2 elevato al numero dello sprite. Se occorrono tutti gli otto sprite, dobbiamo dare 255 come argomento della POKE. Per esempio il comando POKE V+21,6 attiverà gli sprite 1 e 2.

2) Occorre definire l'area di memoria che dà la forma allo sprite. Ogni sprite usa 64 byte di memoria (attualmente solo 63). Nelle locazioni di memoria da 2040 (per lo sprite 0) alla 2047 (per lo sprite 7) dobbiamo scrivere il numero del blocco di 64 byte usato per quel particolare sprite. Qui sono consigliati i numeri da 192 a 199. Per esempio il comando POKE2040,

192 assegnerà i 64 byte partendo da 192*64=12288 allo sprite 0. All'interno degli indirizzi da 12288 a 12350 dobbiamo ora scrivere la forma dello sprite 0. Ogni sprite consiste di una griglia di 24 per 21 punti. Ogni punto di questa griglia può essere acceso o spento. Per completare questo ultimo argomento occorre aggiungere che è possibile assegnare la medesima area di memoria a più di uno sprite, con il risultato di avere più sprite con la stessa forma.

3) Occorre definire il colore dello sprite. I registri dei colori sono gli indirizzi da V+39 (per lo sprite 0) al V+46 (per lo sprite 7) e su questi interveniamo con la POKE per stabilire il colore desiderato con valori da 0 a 15.

4) Occorre definire la posizione orizzontale sullo schermo mediante gli indirizzi V+0 (sprite 0), V+2 (sprite 1) eccetera fino a V+14 (sprite 7). Si può usare un valore da 0 a 255: un valore di 255 posizionerà lo sprite in qualche punto della metà destra dello schermo. Se desiderate muovere uno sprite ulteriormente verso destra dovete posizionare i suoi corrispondenti bit nel registro V+16 (MSB- BIT).

   Per esempio se desiderate porre lo sprite 2 nella posizione orizzontale 265, dovete mettere un 4 con la POKE nella locazione V+16 e un 10 nella locazione V+4.

5) Occorre definire la posizione verticale sullo schermo.

   Questo si fa più facilmente perché sono sufficienti 256 posizioni diverse per coprire l'intera altezza dello schermo: basta porre con la POKE il valore desiderato (da 0 a 255) nella locazione V+1 fino alla V+15. (V+1 per lo sprite 0, V+3 per lo sprite 1, V+5 per lo sprite 2 e così via).

   Nota: la definizione della posizione dello sprite (sia in orizzontale sia in verticale) si riferisce all'angolo superiore sinistro dello sprite. Una volta definiti questi cinque parametri voi dovreste vedere il vostro sprite sullo schermo a meno che i valori X o Y abbiano un valore tale da posizionare lo sprite al di fuori dell'area visibile.

   Aggiungiamo alcune considerazioni riguardanti i registri degli sprite.

6) L'estensione di uno sprite in direzione X può essere raddoppiata posizionando a 1 il corrispondente bit nel registro V+29. Per esempio POKE V+29,8 mostrerà lo sprite 8 con una larghezza doppia nella direzione X.

7) L'estensione di uno sprite in direzione Y può essere raddoppiata posizionando a 1 il corrispondente bit nel registro V+23. Entrambe le larghezze (in orizzontale X e in verticale Y) possono essere riportate alla dimensione normale riposizionando a 0 il bit nel corrispondente registro.

8) È possibile stabilire la priorità degli sprite rispetto allo sfondo. Normalmente se uno sprite si muove sopra lo sfondo (testo, numeri, disegno di un paesaggio) esso appare sovrapposto a tale sfondo. Se desiderate che uno sprite appaia dietro a un oggetto di sfondo dovete posizionare a 1 il corrispondente bit nell'indirizzo V+27. Se due sprite si muovono uno sopra l'altro, quello con il numero più basso ha la priorità su quello con il numero più alto. Lo sprite 0 ha quindi la priorità su tutti.

9) Le collisioni degli sprite tra di loro o con lo sfondo possono essere facilmente individuate. Se uno sprite collide (si sovrappone) con i dati dello sfondo, il bit corrispondente nell'indirizzo V+31 cambia da 0 a 1. Se uno sprite collide con un altro, i bit corrispondenti di entrambi gli sprite coinvolti, nel registro V+30 cambiano da 0 a 1. Entrambi i registri V+30 e V+31 sono cancellati leggendoli con una PEEK.

   Se per esempio muovete lo sprite 0 sullo sprite 1 in modo che si sovrappongano e poi si separino di nuovo, il comando PRINT PEEK(V+30) mostrerà un 3. Questo accade perché i bit 0 e 1 in quel registro sono cambiati a 1 durante la collisione. Se voi digitate di nuovo tale comando verrà visualizzato uno 0. I vostri sprite possono avere un colore uniforme o 3 colori diversi. Per ottenere vari colori dovete attivare il modo multicolore per lo sprite a cui vi riferite. Questo è fatto posizionando il bit corrispondente nel registro V+28.

   I due colori aggiunti (multicolore 1 e multicolore 2) devono essere scritti nei registri V+37 e V+38 (lo stesso per tutti gli sprite). Il colore usato in uno sprite è definito dalla combinazione di 2 bit presi in coppia. Se la coppia dei due bit vale 01, allora verrà usata la funzione multicolore 1, se vale 10 allora verrà impiegato il normale colore degli sprite (registrati da V+39 a V+46), se vale 11 verrà innescata la funzione multicolore 2. Il programma elencato sotto dimostra qualche caratteristica della potenza degli sprite. Otto sprite con forme e colori casuali si muovono rapidamente intorno allo schermo e inoltre il colore dello sfondo cambia di frequente.

```
100 REM 35. ESEMPIO DI SPRITE
110 PRINT"":V=53248
120 POKEV+21,255
130 FOR A=0 TO 7
140 POKE 2040+A,192+A
150 NEXT A
160 FOR B=12288 TO 12798
170 POKE B,255*RND(0)
180 NEXT B
190 FOR C=0 TO 7
200 POKE V+39+C,255*RND(0)
210 NEXT C
```

```
220 POKE V+29,255
230 POKE V+23,255
240 POKE V+28,255
250 POKE V+37,255*RND(0)
260 POKE V+38,255*RND(0)
270 FOR D=0 TO 7
280 POKE V+2*D,255*RND(0)
290 POKE V+2*D+1,255*RND(0)
300 NEXT D
310 POKE V+37,255*RND(0)
320 POKE V+38,255*RND(0)
330 POKEV+33,255*RND(0)
340 GOTO 270

READY.
```

UN GENERATORE DI SPRITE

Il programma listato più avanti vi sarà di notevole aiuto nel creare i vostri sprite.

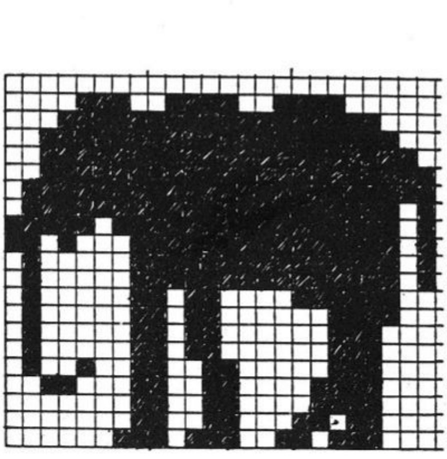
Sullo schermo viene disegnata una cornice entro la quale potrete collocare gli sprite. Basta porre un carattere che non sia uno spazio (per esempio un asterisco) in ogni posizione dove desiderate avere un punto e usare i normali tasti di controllo del cursore per il movimento all'interno della cornice. Per cancellare un punto muovete il cursore sulla posizione desiderata e sovrapponete al carattere esistente uno spazio bianco usando la barra spaziatrice. Non scrivete al di fuori della cornice. Quando avete finito di scrivere i vostri sprite premete RETURN: lo sprite apparirà ora nell'angolo in basso a destra del vostro schermo, nella forma originale. A questo punto avrete le seguenti alternative:

- premendo "H" espanderete lo sprite in orizzontale
- premendo "V" espanderete lo sprite in verticale
- premendo "C" vi riporterete al modo di generazione per effettuare qualche cambiamento
- premendo "M" attiverete il modo multicolore (questo ha senso se avevate disegnato lo sprite con una forma multicolore)
- premendo "O" ritornerete alla visualizzazione originale (non espansa e a colore singolo)
- premendo "Q" visualizzerete i dati relativi al vostro sprite, che potrete utilizzare in una istruzione DATA nel programma dove desiderate inserire lo sprite appena creato.

```
100 REM 36. GENERATORE DI SPRITE
110 PRINT"♥"
120 FORQ=1026TO1049:POKEQ,100:NEXTQ
130 FORQ=55298TO55321:POKEQ,13:NEXTQ
140 FORQ=1065TO1865STEP40:POKEQ,103:NEXTQ
150 FORQ=55337TO56137STEP40:POKEQ,13:NEXTQ
160 FORQ=1090TO1890STEP40:POKEQ,101:NEXTQ
170 FORQ=55362TO56162STEP40:POKEQ,13:NEXTQ
180 FORQ=1906TO1929:POKEQ,99:NEXTQ
190 FORQ=56178TO56211:POKEQ,13:NEXTQ
```

```
195 INPUTA$
200 I=0:Y=0
210 FORX=0TO20
220 FORQ=0TO7
230 IFPEEK(1066+X*40+Q)=32THENA=0:GOTO250
240 A=1
250 I=I+A*(2↑(7-Q))
260 NEXTQ
270 POKE960+Y,I
280 I=0:Y=Y+1
290 FORQ=0TO7
300 IFPEEK(1074+X*40+Q)=32THENA=0:GOTO320
310 A=1
320 I=I+A*(2↑(7-Q))
330 NEXTQ
340 POKE960+Y,I
350 I=0:Y=Y+1
360 FORQ=0TO7
370 IFPEEK(1082+X*40+Q)=32THENA=0:GOTO390
380 A=1
390 I=I+A*(2↑(7-Q))
400 NEXTQ
410 POKE960+Y,I
420 I=0:Y=Y+1
430 NEXTX
440 V=53248
450 POKEV+21,4:POKE2042,15
460 POKEV+23,0:POKEV+29,0
470 POKEV+4,250:POKEV+5,200:REM POSIZIONE
480 POKEV+37,10:POKEV+38,7:POKEV+41,13
490 GETA$:IFA$=""THEN490
500 IFA$="O"THENPOKEV+21,4:POKEV+23,0:POKEV+28,0:POKEV+29,0:POKE2042,15:GOTO490
510 IFA$="M"THENPOKEV+28,4:GOTO490
520 IFA$="H"THENPOKEV+29,4:GOTO490
530 IFA$="V"THENPOKEV+23,4:GOTO490
540 IFA$="C"THENPOKEV+21,0:GOTO180
550 IFA$="Q"THEN570
560 GOTO490
570 PRINT"J"
580 PRINT" NUMERI PER LE ISTRUZIONI DATA :":PRINT
590 FORQ=960TO1022
600 PRINTPEEK(Q);
610 NEXTQ
620 END

READY.
```



0,0,0
14,121,224
31,255,248
63,255,252
63,255,254
63,255,255
127,255,255
127,255,253
251,255,253
209,255,253
65,255,253
65,255,253
65,176,252
65,176,60
65,176,56
65,176,56
73,152,56
49,152,120
1,152,120
1,152,216
3,185,184

Priorità degli sprite

Come detto precedentemente la priorità di uno sprite dipende dal suo numero: lo sprite 0 ha la più alta, lo sprite 7 la più bassa. Il programma listato sotto lo dimostra: otto sprite con la stessa forma, ma con colori diversi sono raffigurati nella stessa posizione e si muovono attorno allo schermo.

Nella linea 120 tutti gli sprite sono attivati. Nella linea 130 tutti gli sprite sono espansi verticalmente e orizzontalmente. Le linee da 140 a 160 definiscono colori diversi per gli otto sprite. Le linee da 170 a 200 definiscono, mediante il comando POKE, la forma memorizzata nelle istruzioni DATA. Le linee da 500 a 520 assegnano questa forma a tutti gli otto sprite. Le linee da 530 a 580 definiscono la posizione degli sprite sullo schermo, che all'inizio è la stessa per tutti. Le linee da 590 a 780 muovono gli sprite in giro per lo schermo per mostrare gli effetti delle priorità.

Le linee da 800 a 830 cambiano i colori degli sprite.

```
100 REM 37. PRIORITA' DEGLI SPRITES
110 V=53248
120 POKEV+21,255
130 POKEV+23,255:POKEV+29,255
140 FORX=0TO7
150 POKEV+39+X,X+7
160 NEXTX
170 FORX=0TO62
180 READA
190 POKE12288+X,A
200 NEXTX
300 DATA 0,24,0,0,60,0,0,126,0,0,255,0
310 DATA 1,255,128,3,255,192,7,255,224
320 DATA 15,255,240,31,255,248,63,255,252
330 DATA 127,255,254,63,255,252,31,255
340 DATA 248,15,255,240,7,255,224,3,255
350 DATA 192,1,255,128,0,255,0,0,126,0
360 DATA 0,60,0,0,24,0
500 POKE2040,192:POKE2041,192:POKE2042,192
510 POKE2043,192:POKE2044,192:POKE2045,192
520 POKE2046,192:POKE2047,192
530 FORX=0TO14STEP2
540 POKEV+X,100
550 NEXTX
560 FORY=1TO15STEP2
570 POKEV+Y,80
580 NEXTY
590 FORX=0TO14STEP2
600 FORZ=0TO100
610 POKEV+X,100+Z
620 NEXTZ
630 NEXTX
640 FORY=1TO15STEP2
650 FORZ=0TO100
660 POKEV+Y,80+Z
670 NEXTZ
680 NEXTY
690 FORX=0TO14STEP2
700 FORZ=1TO10*X
710 POKEV+X,200-Z
720 NEXTZ
730 NEXTX
740 FORY=1TO15STEP2
750 FORZ=1TO5*Y
760 POKEV+Y,180-Z
770 NEXTZ
```

```
780 NEXTY
790 Z=8
800 POKEV+39+Z,15*RND(0)
810 Z=Z-1:IFZ=0THENZ=8
820 FORD=1TO50:NEXTD
830 GOTO800

READY.
```

Segnalazione di collisione

Come descritto precedentemente la video-chip del COMMODORE 64 ha dei registri che indicano se gli sprite si sovrappongono tra di loro o con parti dello sfondo. Questa caratteristica è di aiuto nel programmare giochi. Per esempio in un gioco dove un oggetto (sprite) deve muoversi attraverso un tunnel (sfondo), noi controlliamo periodicamente il registro opportuno e, se esso indica una collisione tra lo sprite e lo sfondo programmiamo una specie di esplosione e diminuiamo il punteggio. Una cosa da sottolineare è che, a collisione avvenuta, i bit nel registro di collisione restano posizionati anche se non c'è più sovrapposizione: il leggere il registro di collisione li cancella.

Il programma elencato sotto mostra come il registro di collisione lavora. Uno sprite si muove sopra (o sotto) un altro per lo stesso tempo viene visualizzato il registro di collisione (quello che indica la collisione fra due sprite). Non appena i due sprite sovrappongono un bit, il contenuto cambia da 0 a 3. Il 3 è dovuto al fatto che lo sprite 0 e lo sprite 1 collidono (00000011=3). In questo modo è possibile stabilire quali sprite sono venuti a collisione e con quali altri. Un contenuto di 192 indica per esempio che la collisione è avvenuta tra gli sprite 7 e 0. Oltre al registro che indica la collisidone tra sprites ce ne è un altro che indica la collisione degli sprite con lo sfondo.

```
100 REM 38. SEGNALAZIONE DI COLLISIONI
110 PRINT"♥":V=53248
120 POKEV+21,3
130 POKEV+23,3:POKEV+29,3
140 POKEV+39,7:POKEV+40,8
150 POKE2040,192:POKE2041,192
160 POKEV,120:POKEV+2,200
170 POKEV+1,100:POKEV+3,100
180 FORX=0TO62
190 READA:POKE12288+X,A
200 NEXTX
210 FORX=0TO150
220 POKEV+2,200-X
230 PRINT"■■REGISTRO DELLE COLLISIONI :";PEEK(V+30)
240 NEXTX
250 GOTO210
300 DATA 0,24,0,0,60,0,0,126,0,0,255,0
310 DATA 1,255,128,3,255,192,7,255,224
320 DATA 15,255,240,31,255,248,63,255,252
330 DATA 127,255,254,63,255,252,31,255
340 DATA 248,15,255,240,7,255,224,3,255
350 DATA 192,1,255,128,0,255,0,0,126,0
360 DATA 0,60,0,0,24,0

READY.
```

Il passo successivo dopo aver individuato una collisione è di programmare qualche genere di esplosione e questo è mostrato nel programma che segue. Qui usiamo un'area di memoria separata per definire la forma del secondo sprite perché desideriamo cambiare questa forma dopo l'individuazione di una collisione. Osserviamo la linea 205: questa linea legge il contenuto del registro di collisione e cancella anche la segnalazione di eventuali precedenti collisioni. La linea 230 controlla la collisione degli sprite 0 e 1. Le linee da 400 a 420 cambiano il color dello sprite 1. Le linee da 400 a 500 cambiano la forma dello sprite 1.

```
100 REM 39. EXPLOSIONE DOPO UMA COLLISIONE
110 PRINT"J":V=53248
120 POKEV+21,3
130 POKEV+23,3:POKEV+29,3
140 POKEV+39,7:POKEV+40,8
150 POKE2040,192:POKE2041,193
160 POKEV,120:POKEV+2,200
170 POKEV+1,100:POKEV+3,100
180 FORX=0TO62
190 READA:POKE12288+X,A:POKE12288+64+X,A
200 NEXTX
205 Q=PEEK(V+30)
210 X=200
220 POKEV+2,X
230 IFPEEK(V+30)=3THEN400
240 X=X-1
250 GOTO220
300 DATA 0,24,0,0,60,0,0,126,0,0,255,0
310 DATA 1,255,128,3,255,192,7,255,224
320 DATA 15,255,240,31,255,248,63,255,252
330 DATA 127,255,254,63,255,252,31,255
340 DATA 248,15,255,240,7,255,224,3,255
350 DATA 192,1,255,128,0,255,0,0,126,0
360 DATA 0,60,0,0,24,0
400 FORX=1TO30
410 POKEV+40,15*RND(0)
420 NEXTX
430 POKEV+40,8
440 FORX=0TO54STEP2
450 POKE12288+64+X,PEEK(12288+64+X)AND170
460 NEXTX
470 FORX=0TO54
480 POKE12288+64+X,0
490 FORD=1TO20:NEXTD
500 NEXTX

READY.
```

Come abbiamo appreso in precedenza, possiamo assegnare a uno sprite una certa area di memoria dove la forma di tale sprite è definita. Possiamo per esempio avere otto diverse forme memorizzate in otto diverse locazioni di memoria e assegnare ogni forma a uno sprite in modo di averne otto diversi sullo schermo. Un'altra possibilità è di avere per esempio due forme diverse e assegnare a quattro sprite la prima forma e agli altri quattro la seconda; otterremo così un totale di otto sprite con due tipi di forme. Un'altra possibilità consiste nell'usare la medesima forma per tutti gli otto sprite. Se

desideriamo creare un oggetto in movimento, per esempio un omino che passeggia, ci servirà più di una forma per sprite, ma con il COMMODORE 64 lo si può fare facilmente.

Dobbiamo creare due o più forme diverse in memoria e poi assegnarle, una dopo l'altra, allo sprite. Questo crea un effetto simile a quello di un film comico. Se poi creiamo dieci forme differenti e le selezioniamo una per una potremo creare l'illusione più realistica di un oggetto in movimento, ad esempio un uomo che corre. Nel medesimo tempo noi selezioneremo un'altra forma perché desidereremo muovere lo sprite sullo schermo cambiando il registro della posizione X o della posizione Y. Il programma listato più avanti usa la tecnica appena descritta. Ci sono tre diverse forme memorizzate nelle istruzioni DATA delle linee 300 - 360, 370 - 430, 440 - 500.

Nella linea 1010 viene selezionata una delle tre forme e nella linea 1015 viene cambiata la posizione X dello sprite. Il programma crea quindi l'illusione di un barile che rotola.

Gli altri due programmi elencati usano le stesse tecniche: il primo simula un cavallo che corre, il secondo un omino multicolore. Come già detto, il movimento appare più naturale aggiungendo ulteriori immagini.

```
100 REM 40. ESEMPIO DI SPRITE
110 V=53248
120 POKEV+21,1
140 POKEV+39,50
160 POKEV+1,200
170 FORX=0TO62
180 READ A
190 POKE 12288+X,A
200 NEXT X
210 FOR X=0 TO 62
220 READ A
230 POKE 12352+X,A
240 NEXT X
250 FOR X=0 TO 62
260 READ A
270 POKE 12416+X,A
280 NEXT X
300 DATA 255,255,255,255,0,1,255,0,1
310 DATA 255,0,1,255,0,1,255,0,1
320 DATA 255,0,1,255,0,1,255,0,1
330 DATA 255,0,1,255,0,1,255,0,1
340 DATA 255,0,1,255,0,1,255,0,1
350 DATA 255,0,1,255,0,1,255,0,1
360 DATA 255,0,1,255,0,1,255,255,255
370 DATA 255,255,255,128,255,1,128,255,1
380 DATA 128,255,1,128,255,1,128,255,1
390 DATA 128,255,1,128,255,1,128,255,1
400 DATA 128,255,1,128,255,1,128,255,1
410 DATA 128,255,1,128,255,1,128,255,1
420 DATA 128,255,1,128,255,1,128,255,1
430 DATA 128,255,1,128,255,1,255,255,255
440 DATA 255,255,255,128,0,255,128,0,255
450 DATA 128,0,255,128,0,255,128,0,255
460 DATA 128,0,255,128,0,255,128,0,255
470 DATA 128,0,255,128,0,255,128,0,255
480 DATA 128,0,255,128,0,255,128,0,255
490 DATA 128,0,255,128,0,255,128,0,255
500 DATA 128,0,255,128,0,255,255,255,255
```

```
990 POKEV+23,1:POKEV+29,1
1000 FOR X=0 TO 2
1010 POKE 2040,192+X
1015 POKE V,Q:Q=Q+5:IF Q>250 THEN Q=20
1018 FOR D=1 TO 50:NEXT D
1020 NEXT X
1030 GOTO 1000

READY.
```

```
100 REM 41. DUE IMMAGINI DIVERSE
110 V=53248
120 POKEV+21,1
130 POKEV+23,1:POKEV+29,1
140 POKEV+39,50
160 POKEV+1,200
170 FORX=0TO62
180 READA
190 POKE12288+X,A
200 NEXT X
210 FORX=0TO62
220 READA
230 POKE12352+X,A
240 NEXTX
300 DATA 0,0,0,0,0,0,0,0,144
310 DATA 0,0,96,0,0,240,0,1,248
320 DATA 131,255,252,71,255,238,47,255,196
330 DATA 31,255,128,15,255,128,15,255,128
340 DATA 15,255,128,20,1,64,20,1,64
350 DATA 20,1,64,36,1,32,72,0,144
360 DATA 144,0,72,144,0,72,0,0,0
400 DATA 0,0,0,0,0,0,0,0,144
410 DATA 0,0,96,0,0,240,0,1,248
420 DATA 3,255,252,7,255,238,15,255,196
430 DATA 255,255,128,15,255,128,15,255,128
440 DATA 15,255,128,20,1,64,20,1,64
450 DATA 20,1,64,20,1,64,18,2,80
460 DATA 9,4,128,9,4,128,0,0,0
500 POKE 2040,192
510 POKE V,Q:Q=Q+3:IF Q>250 THEN Q=20
515 FOR D=1 TO 50:NEXT D
518 POKE V,Q:Q=Q+3
520 POKE 2040,193
530 FOR D=1 TO 50:NEXT D
540 GOTO 500

READY.
```

```
100 REM 42. SPRITES
110 V=53248
120 POKE V+21,1:REM VIENE ATTIVATO LO SPRITE 1
130 POKE V+39,7:REM COLORE
140 POKE V+28,1:REM VIENE ATTIVATO IL MODO MULTI-COLORE
150 POKE V+37,2:REM MULTI-COLORE 1
160 POKE V+38,3:REM MULTI-COLORE 2
170 POKE V+29,1:REM ESPANSIONE IN ORIZZONTALE
180 POKE V+23,1:REM ESPANSIONE IN VERTICALE
```

```
190 POKE V+1,100:REM POSIZIONE Y
200 REM VIENE DEFINITA LA FORMA 1
210 FOR X=0 TO 62
220 READ A:POKE 12288+X,A
230 NEXT X
240 REM VIENE DEFINITA LA FORMA 2
250 FOR X=0 TO 62
260 READ A:POKE 12352+X,A
270 NEXT X
280 REM ISTRUZIONI DATA PER LA FORMA 1
290 DATA 0,84,0,0,84,0,0,84,0
300 DATA 0,84,0,0,32,0,0,32,0
310 DATA 0,168,0,0,168,0,2,34,0
320 DATA 2,34,0,8,32,128,8,32,128
330 DATA 0,32,0,0,48,0,0,48,0
340 DATA 0,48,0,0,48,0,0,48,0
350 DATA 0,48,0,0,60,0,0,60,0
360 REM ISTRUZIONI DATA PER LA FORMA 2
370 DATA 0,84,0,0,84,0,0,84,128
380 DATA 0,84,128,0,34,0,0,34,0
390 DATA 0,168,0,0,168,0,2,32,0
400 DATA 2,32,0,8,32,0,8,32,0
410 DATA 0,32,0,0,60,0,0,60,0
420 DATA 0,51,48,0,51,48,0,48,192
430 DATA 0,48,192,0,60,0,0,60,0
440 POKE V+0,30
450 REM VIENE SELEZIONATA LA FORMA 1
460 POKE 2040,192
470 REM CAMBIO DELLA LOCAZIONE X
480 POKE V+0,Q:Q=Q+5:IF Q>250 THEN Q=30
490 REM RITARDO
500 FOR D=1 TO 50:NEXT D
510 REM VIENE SELEZIONATA LA FORMA 2
520 POKE 2040,193
530 REM CAMBIO DELLA LOCAZIONE X
540 POKE V+0,Q:Q=Q+5
550 REM RITARDO
560 FOR D=1 TO 100:NEXT D
570 REM RIPETIZIONE
580 GOTO 450

READY.
```

SUONO

Il vostro COMMODORE 64 contiene all'interno uno dei circuiti a tecnologia più avanzata per il trattamento del suono.

In questo capitolo ci proponiamo di comprendere e utilizzare questa chip. Il circuito del suono detto anche "SOUND INTERFACE DEVICE" o abbreviato SID (che significa dispositivo di interfaccia del suono) è situato in memoria alla locazione $D400, cioè 54272. Noi assegneremo tale valore alla variabile S, cioè porremo S=54272.

Il SID comprende:

3 voci indipendenti

4 forme d'onda (rumore, pulsazione, a dente di sega, a triangolo)

3 modulatori di ampiezza

3 generatori di inviluppo

1 sincronizzatore di oscillazioni

1 modulatore ad anello

Ci sono 7 registri per il controllo di ognuno dei 3 oscillatori più 4 registri aggiuntivi per il controllo del filtro e del volume. Due altri registri vi permettono di leggere certi parametri dell'oscillatore 3, che può essere usato per influenzare gli altri oscillatori. Oltre ai registri menzionati il SID ha due convertitori A/D che vi permettono di leggere la posizione di un potenziometro (per esempio una paddle).

I registri della chip del suono devono essere cancellati con il comando "POKE registro, O" all'inizio di ogni programma per escludere suoni eventualmente esistenti.

Per attivare il canale del suono dovete procedere nel seguente modo:

1) caricate il registro S+24 con il volume (da 0 a 15)

2) caricate i registri per definire gli ADSR (S+5 e S+6 per l'oscillatore 1, S+12 e S+13 per l'oscillatore 2, S+19 e S+20 per l'oscillatore 3). Questi

registri definiscono come un suono inizia (aumentando di volume) e termina (diminuendo di volume)

3) caricate i registri per la definizione di frequenza (S+0 e S+1 per l'oscillatore 1, S+7 e S+8 per l'oscillatore 2, S+14 e S+15 per l'oscillatore 3)

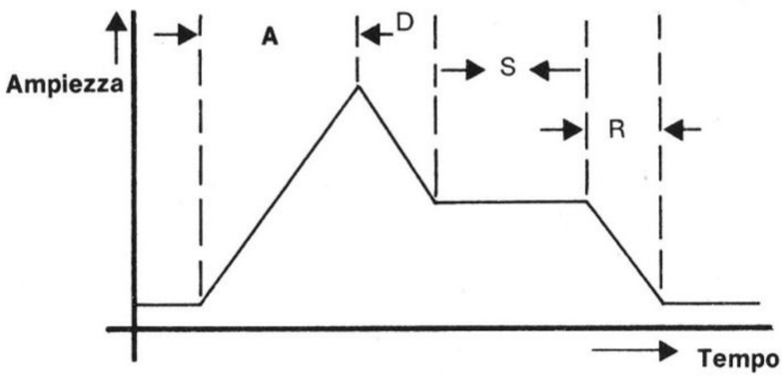
4) se desiderate selezionare la forma d'onda della pulsazione, caricate i registri con l'ampiezza di pulsazione (S+2 e S+3 per l'oscillatore 1, S+9 e S+10 per l'oscillatore 2, S+16 e S+17 per l'oscillatore 3)

5) selezionate la forma d'onda e generate il tono (registri S+4, S+11, S+18 rispettivamente)

6) definite il ciclo di ritardo

7) iniziate il rilascio del tono riposizionando il bit che lo ha generato (registri S+4, S+11, S+18 rispettivamente).

Il ciclo ATTACK-DECAY-SUSTAIN-RELEASE relativo a un tono è mostrato nel diagramma seguente:



Descrizione dei registri del SID.

L'indirizzo di base del SID è il $D400, in decimale 54272: per indirizzare un registro dovete aggiungere il suo numero all'indirizzo di base. Alcuni registri controllano più di un parametro, altri sono usati solo parzialmente.

Registro 0:
questo registro contiene gli otto bit più bassi della frequenza dell'oscillatore 1. In questo indirizzo potete scrivere un valore da 0 a 255 e più alto è il valore, più alta sarà la frequenza dell'oscillatore 1.

Registro 1:
contiene gli otto bit più alti di frequenza dell'oscillatore 1, con valori da 0 a 255.

Registro 2:
contiene gli otto bit più bassi dell'ampiezza di pulsazione, con valori da 0 a 255.

Registro 3:
contiene i quattro bit più alti dell'ampiezza di pulsazione con valori da 0 a 15 (gli otto bit superiori di questo registro non sono usati). I registri 2 e 3 combinati definiscono il rapporto tra picchi e spazi della forma d'onda pulsante.

Se scrivete 15 nel registro 3 e 255 nel registro 2 otterrete uno spazio (o pausa), che rappresenta una uscita continua ovvero un'assenza di pulsazioni. Se scrivete 8 nel registro 3 e 0 nel registro 2 allora il rapporto tra picchi e spazi (pulsazioni e pause) è 1 a 1, che rappresenta una tipica onda quadra. Se selezionate la forma d'onda pulsante usando il registro 4, è necessario assegnare i valori solamente ai registri 2 e 3.

Registro 4:
è il registro di controllo della voce 1. Ogni bit in questo registro ha un significato diverso. Il bit più alto (7) posizionato a 1 attiva il generatore di rumore della voce 1. Il bit 6 posizionato a 1 seleziona una forma d'onda pulsante e in questo caso i registri 2 e 3 devono contenere i valori che definiscono il rapporto pulsazione/pausa. Il bit 5 posto a 1 seleziona una forma d'onda a dente di sega. Il bit 4 posto a 1 seleziona una forma d'onda triangolare.Il bit 3 posto a 1 chiude l'oscillatore 1 e questo vi permette di sincronizzare l'oscillatore 1 con un evento esterno. Normalmente questo bit non è utilizzato e viene lasciato a 0. Il bit 2 posto a 1 vi permette di agire con la voce 3 sulla voce 1. Se desiderate utilizzare questa possibilità provate a posizionare la voce 1 a forma d'onda triangolare e a definire una frequenza per la voce 3: quello che voi udrete non sarà più una forma d'onda a triangolo, ma una combinazione della voce 1 e della voce 3.

Il bit 1 posto a 1 vi permette di sincronizzare la voce 1 con la voce 3 creando interessanti effetti sonori. Il bit 0 posto a 1 fa partire il generatore di

inviluppo per la voce 1; questo significa che il volume della voce 1 cresce da 0 al suo valore di picco entro un tempo definito dal registro 5 (parte di ATTACK) e ritorna al livello di SUSTAIN (definito nel registro 6) entro un tempo definito ancora nel registro 5. Se il bit 0 è posto di nuovo a 0 il volume della voce 1 va dal livello SUSTAIN a 0 entro un tempo definito nel registro 6 (parte di RELEASE).

Registro 5:
questo registro controlla due cose: il tempo di ATTACK e il tempo di DECAY. I quattro bit superiori del registro (bit da 4 a 7) definiscono il tempo di ATTACK, cioè il tempo in cui il volume va da zero al suo valore massimo. I quattro bit inferiori del registro (i bit da 0 a 3) definiscono il tempo entro cui il volume va dal suo massimo al livello di SUSTAIN selezionato (vedi registro 6).

Registro 6:
anche questo registro controlla due elementi, il livello di SUSTAIN e il tempo di RELEASE. I quattro bit superiori del registro (bit da 4 a 7) controllano il livello (SUSTAIN) al quale si assesta il volume partendo dal suo picco entro il tempo di DECAY. Sono possibili 16 posizionamenti. Il porre tutti e quattro i bit a 1 significa un livello di SUSTAIN al massimo volume, tutti e quattro i bit a 0 significa livello di SUSTAIN nullo. Il numero binario 1000, che equivale al decimale 128 significa un livello di SUSTAIN pari alla metà del volume di picco (poiché 128 è la metà superiore del byte).

I quattro bit inferiori del registro definiscono il tempo entro cui il volume va dal livello di SUSTAIN a 0. Il ciclo di RELEASE inizia non appena il bit 0 del registro 4 è posto di nuovo a 0. Alcuni valori dei tempi di RELEASE e di DECAY sono: i valori 0,2,4,8,10,12,15 conducono rispettivamente a 6,48,114,300,1.5,3 e 24 millisecondi e i corrispondenti tempi di ATTACK sono: 2,16,38,100,500 millisecondi, 1 e 8 secondi.

Registri da 7 a 13:
questi registri controllano la voce 2 così come i registri da 0 a 6 controllano la voce 1. Le sole differenze sono:

- se selezionate SYNC per la voce 2 (bit 1 del registro 11), la voce 2 sarà sincronizzata con la voce 1;
- se selezionate RINGMOD per la voce 2 (bit 2 del registro 11), allora la forma d'onda triangolare della voce 2 sarà sostituita da una combinazione modulata ad anello della voce 2 con la voce 1.

Registri da 14 a 20:

questi registri controllano la voce 3 così come i registri da 0 a 6 controllano la voce 1. Le sole differenze sono:

- se selezionate SYNC per la voce 3 (bit 1 del registro 18, allora la voce 3 sarà sincronizzata con la voce 2;

- se selezionate RINGMOD per la voce 3 (bit 2 del registro 18), allora la forma d'onda triangolare della voce 3 sarà sostituita da una combinazione modulata ad anello della voce 3 con la voce 2.

Registro 21:

I tre bit più bassi di questo registro (bit 0,1,2) sono riservati alla parte inferiore della frequenza di filtro. È consigliabile usare valori da 0 a 7.

Registro 22:

questo registro di otto bit serve per la parte superiore della frequenza di filtro (chiamata anche frequenza di taglio).

I registari 21 e 22 insieme vi permettono di definire la frequenze di filtro da 30HZ a 12HZ.

Registro 23:

i quattro bit superiori di questo registro (bit da 0 a 3) servono per controllare la risonanza del filtro. Un valore di 0 significa assenza di risonanza, quattro 1 significano un filtro molto ristretto che lascia passare solo una certa frequenza e attenua fortemente le altre. I quattrao bit più bassi di questo registro definiscono quale voce deve essere filtrata.

Se il bit 3 è posto a 1, un impulso audio esterno è mandato attaraverso il filtro. Se il bit 2 è posto a 1 viene filtrata la voce 3, se il bit 1 vale 1 viene filtrata la voce 2 e così per la voce 1 se il bit 0 vale 1.

Registro 24:

i quattro bit superiori di questo registro hanno le seguenti funzioni: se il bit 7 vale 1 allora la voce 3 viene eliminata, il che può esservi utile se tale voce serve solo di modulazione per le altre, ma non desiderate udirla.

Se il bit 6 vale 1 viene selezionato un filtro passa - alto.
Se il bit 5 vale 1 viene selezionato un filtro passa - banda.
Se il bit 4 vale 1 viene selezionato un filtro passa - basso.

In dipendenza dal tipo di filtro selezionato tutte le frequenze al di sotto (nel caso di filtro passa - alto) o al di sopra (nel caso di filtro passa - basso) della frequenza filtrata verranno soppresse. Se avete selezionato un filtro passa - banda, allora tutte le frequenze al di sopra e al di sotto della frequenza filtrata verranno eliminate ma non in modo così accentuato come nel caso di filtro passa - alto o passa - basso. Se selezionate un filtro passa - alto e un filtro passa - basso contemporaneamente, avrete per risultato un filtro a barriera di banda che lascia passare tutte le frequenze tranne una piccola porzione intorno alla frequenza filtrata.

Registro 25:

questo registro vi permette di leggere un valore corrispondente alla posizione di un potenziometro esterno, per esempio una paddle, infilato sulla connessione chiamata POTX.

Registro 26:

come per il registro 25, ma è valido per la connessione POTY. I due collegamenti appena ricordati sono connessi alle porte del joystick.

Registro 27:

questo registro può essere impiegato come un generatore di numeri casuali, se la voce 3 è posizionata sul modo "rumore".

Questo vi permette di leggere i quattro bit superiori d uscita della voce 3. Se la voce 3 è posizionata a forma d'onda a dente di sega, il registro 27 contiene numeri che si incrementano da 0 a 255. Se alla voce 3 è assegnata una forma d'onda triangolare, il registro 27 contiene numeri che si incrementano da 0 a 255, quindi diminuiscono fino a 0. Se alla voce 3 è assegnata infine una forma d'onda pulsante, il registro 27 contiene numeri che saltano da 0 a 255.

Registro 28:

questo registro è simile al registro 27, con la differenza che in questo caso l'uscita del generatore di inviluppo della voce 3 può essere letta. Influenzando le altre voci mediante i registri 27 e 28 si ottengono effettti sonori speciali.

Più sotto vengono elencati due programmi: il primo vi permette di inserire direttamente dei numeri (binari) nei registri della chip del suono e voi potete usare i tasti di controllo del cursore per passare ad un altro registro. Il secondo vi mostra l'influenza dell'ampiezza di pulsazione sul suono quando usate una forma d'onda pulsante.

```
99 REM 43.
100 PRINT"ℸ":S=54272:Z=0
110 PRINT"VOCE 1 FREQ LO ";"[.... ....]"
120 PRINT"       FREQ HI ";"[.... ....]"
130 PRINT"       PW   LO ";"[.... ....]"
140 PRINT"       PW   HI ";"[**** ....]"
150 PRINT"       CONTROL ";"[.... ....]"
160 PRINT"       ATT/DEC ";"[.... ....]"
170 PRINT"       SUS/REL ";"[.... ....]"
180 PRINT"VOCE 2 FREQ LO ";"[.... ....]"
190 PRINT"       FREQ HI ";"[.... ....]"
200 PRINT"       PW   LO ";"[.... ....]"
210 PRINT"       PW   HI ";"[**** ....]"
220 PRINT"       CONTROL ";"[.... ....]"
230 PRINT"       ATT/DEC ";"[.... ....]"
240 PRINT"       SUS/REL ";"[.... ....]"
250 PRINT"VOCE 3 FREQ LO ";"[.... ....]"
260 PRINT"       FREQ HI ";"[.... ....]"
270 PRINT"       PW   LO ";"[.... ....]"
280 PRINT"       PW   HI ";"[**** ....]"
290 PRINT"       CONTROL ";"[.... ....]"
300 PRINT"       ATT/DEC ";"[.... ....]"
310 PRINT"       SUS/REL ";"[.... ....]"
320 PRINT"FILTRO FREQ LO ";"[**** *...]"
330 PRINT"       FREQ HI ";"[.... ....]"
340 PRINT"       RES/FIL ";"[.... ....]"
350 PRINT"       MOD/VOL ";"[.... ....]";
360 PRINT"▣";
370 PRINT "▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶";
380 FOR C=1 TO 4
385 POKE 204,0
390 GET A$(C):IF A$(C)="" THEN 390
391 IF PEEK(207) THEN 391
392 POKE 204,1
393 IF A$(C)="▣"THEN 540
394 IF A$(C)="ℸ" THEN Z=Z-1:PRINT"ℸℸ":GOTO 396
395 GOTO 398
396 IF Z<0 THEN Z=24:PRINT"▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣";
397 GOTO 370
398 IF A$(C)="1" OR A$(C)="0" THEN 400
399 GOTO 385
400 PRINTA$(C);
410 NEXT C
420 PRINT"▶";
430 FOR C=1 TO 4
432 POKE 204,0
440 GET B$(C):IF B$(C)="" THEN 440
442 IF PEEK(207) THEN 442
444 POKE 204,1
446 IF B$(C)="1" OR B$(C)="0" THEN 450
448 GOTO 432
450 PRINTB$(C);
460 NEXT C
470 T=0
480 FOR C=0 TO 3
490 IF VAL(A$(C+1))=1 THEN T=T+2↑(7-C)
500 IF VAL(B$(C+1))=1 THEN T=T+2↑(3-C)
510 NEXT C
512 N$=STR$(T):IF LEN(N$)=4 THEN 520
514 IF LEN(N$)=3 THEN 518
516 N$=" "+N$
518 N$=" "+N$
520 PRINT"▶";S+Z;"=";N$;
530 POKE S+Z,T
540 Z=Z+1:IF Z=25 THEN Z=0:GOTO 360
550 PRINT:GOTO 370

READY.
```

```
99 REM 44.
100 PRINT"♥"
110 PRINT"DIMOSTRAZIONE DI AMPIEZZA DI PULSAZIONI"
120 PRINT:PRINT"'*' SIGNIFICA PULSAZIONE ALTA"
130 PRINT"'.' SIGNIFICA PULSAZIONE BASSA"
140 POKE54296,15
150 POKE54272,16:POKE54273,16
160 POKE54276,64+1
170 FOR A=0 TO 15
180 FOR B=0 TO 255 STEP 17
190 PRINT"SQQQQQQQQQQQQ]";
200 PRINT"....................................."
210 PRINT"SQQQQQQQQQ]";
220 FOR X=1 TO (256*A+B)/110
230 PRINT"*";
240 NEXT X
250 POKE 54275,A
260 POKE 54274,B
270 A$=STR$(A):A$=A$+"  "
280 B$=STR$(B):B$=B$+"  "
290 PRINT"SQQQQQQQQQQQQQQQ]REGISTRO PW HI =";A$
300 PRINT"SQQQQQQQQQQQQQQQQ]REGISTRO PW LO =";B$
310 NEXT B
320 NEXT A

READY.
```

JOYSTICK E PADDLE

Alla destra del vostro COMMODORE 64 potete trovare due connettori a 9 attacchi chiamati porta 1 e porta 2: ad entrambe si possono attaccare un joystick (leva di comando) o una paddle (paletta). Alla porta di controllo si può anche attaccare una penna luminosa. La differenza tra un joystick e una paddle è che con un joystick voi potete comunicare la direzione di un movimento, mentre con una paddle voi assegnate un valore assoluto. Per esempio se desiderate muovere uno sprite orizzontalmente (nella direzione X) sullo schermo, dovete fare un programma in questo modo:

M: leggere la porta del joystick

controllare la direzione orizzontale

decrementare/incrementare il contatore

scrivere i contenuti del contatore nel registro X dello sprite

saltare indietro a M

Con una paddle il programma diventa:

M: leggere la porta della paddle (convertitore A/D)

scrivere il valore nel registro dello sprite

saltare indietro a M

Un joystick consiste attualmente di quattro interruttori: su, giù, sinistra, destra. A seconda della direzione in cui muovete il joystick uno o due degli interruttori sono chiusi, dando origine a un numero binario che può essere letto dalla porta. Una paddle consiste in un potenziometro che ha una certa resistenza che dipende dalla sua posizione. Questa resistenza è trasformata in un numero da 0 a 255 da un convertitore analogico-digitale (A/D) inserito nel vostro COMMODORE 64.

Sia il joystick sia la paddle hanno anche un altro interruttore costituito da un bottone per fare fuoco.

La posizione del joystick può essere letta dai comandi:

PEEK (56321) per il joystick 1
PEEK (56320) per il joystick 2

A seconda della posizione del joystick questi comandi restituiranno i seguenti valori:

joystick in posizione neutra		: 127
su	N	: 126
giù	S	: 125
sinistra	W	: 123
destra	E	: 119
	NW	: 122
	NE	: 118
	SW	: 121
	SE	: 117

Quando il bottone del fuoco è premuto leggete i valori soprascritti meno 16.

Le posizioni delle paddle possono essere lette dai registri 54297 e 54298 se state usando le paddle ATARI; entrambe sono infilate nello stesso connettore e questo è inserito nella porta di controllo 1.

Il programma listato sotto vi permette di muovere uno sprite sia orizzontalmente sia verticalmente usando 2 paddle e imitando il cosiddetto "MOUSE" (topo) che vi permette di muovere il cursore in ogni locazione dello schermo. Nelle linee 190 e 200 viene letta la posizione della paddle e questo valore è utilizzato per la posizione orizzontale e verticale dello sprite.

```
100 REM 45. USO DELLE PADDLES
110 PRINT"J"
120 V=53248
130 POKEV+21,1
140 POKEV+39,8
150 POKE2040,192
160 FORX=0TO62
170 READA:POKE12288+X,A
180 NEXTX
190 POKEV,PEEK(54297)
200 POKEV+1,PEEK(54298)
210 GOTO190
300 DATA 0,24,0,0,60,0,0,126,0,0,255,0
310 DATA 1,255,128,3,255,192,7,255,224
320 DATA 15,255,240,31,255,248,63,255,252
330 DATA 127,255,254,63,255,252,31,255
340 DATA 248,15,255,240,7,255,224,3,255
350 DATA 192,1,255,128,0,255,0,0,126,0
360 DATA 0,60,0,0,24,0

READY.
```

Le paddle servono anche per influenzare il suono. Il programma seguente ne è un esempio. La chip del SID crea un suono sulla voce 1, influenzata dalla voce 3. La frequenza della voce 1 è controllata da una paddle e quella della voce 3 dall'altra.

In questo modo dovreste essere capaci di generare svariati tipi di suoni diversi.

```
100 REM 46. MUSICA CON LE PADDLES
110 POKE 54296,128+15
120 POKE 54277,128+8
130 POKE 54278,128+8
140 POKE 54291,128+8
150 POKE 54292,240
160 POKE 54290,16+1
170 POKE 54276,16+4+0
180 POKE 54273,PEEK(54297)/4
190 POKE 54287,PEEK(54298)/8
200 POKE 54276,16+4+1
210 FORD=1TO80:NEXTD
220 GOTO 170

READY.
```

La prima parte del libro contiene una serie di programmi in BASIC Commodore pronti per essere introdotti nel vostro calcolatore ed eseguiti con il comando RUN.
La seconda parte vuole insegnarvi ad utilizzare a pieno la notevole potenza del Commodore 64, attraverso il colore, la grafica ad alta risoluzione, gli sprite, il suono, i joystick e le paddle.
Quando sarete arrivati alla fine del libro sarete capaci di scrivere da soli programmi sofisticati per giochi movimentati e divertenti.

L. 24.000 Cod. 408D ISBN 88-7056-155-0